

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra kybernetiky a biomedicínského
inženýrství

Detekční systém s identifikací osob z
obrazového signálu analýzy prstu

Detection System with Identification of
Person by Image Recognition of Finger
Analysis

2015

Bc. Miroslav Praženka

Zadání

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra kybernetiky a biomedicínského inženýrství

Zadání diplomové práce

Student: **Bc. Miroslav Praženka**
Studijní program: N2649 Elektrotechnika
Studijní obor: 3901T009 Biomedicínské inženýrství
Téma: **Detekční systém s identifikací osob z obrazového signálu analýzy prstu**
Detection System with Identification of Person by Image Recognition of Fingure Analyses

Zásady pro vypracování:

1. Rozbor problematiky metod identifikace obrazového signálu analýzou prstu.
2. Návrh systému pro identifikace obrazového signálu analýzou prstu.
3. Realizace softwaru pro identifikaci obrazového signálu analýzou prstu.
4. Verifikace naměřených výsledků s teoretickými předpoklady.
5. Zhodnocení dosažených výsledků.

Seznam doporučené odborné literatury:

- [1] PHILIPS, Dwayne. *Image processing in C*. Lawrence Kansas: R & D Publications, 1994. ISBN 0-13-104548-2. 2. digitalizované vydání v roce 2000.
[2] HLAVÁČ, Václav a Miloš SEDLÁČEK. *Zpracování signálů a obrazů*. 2.přepřac.vyd. Praha: Nakladatelství ČVUT, 2007. 255 s. ISBN 978-80-01-03110-0.
[3] BOVIK, Alan C. *Handbook of Image and Video Processing*. 1.vyd. San Diego: Academic Press, c2000. 891 s. ISBN 0-12-119790-5.
[4] HALOUNOVÁ, Lena. *Zpracování obrazových dat*. 1. vyd. Praha: Česká technika - nakladatelství ČVUT, 2009. 102 s. ISBN 978-80-01-04253-3.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Zdeněk Macháček, Ph.D.**

Datum zadání: 01.09.2014

Datum odevzdání: 07.05.2015

doc. Ing. Jiří Koziorek, Ph.D.
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení studenta

Prohlašuji, že jsem tuto diplomovou práci, *Detekční systém s identifikací osob z obrazového signálu analýzy prstu*, vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě r. 2015



.....
vlastnoruční podpis autora

Poděkování

Rád bych tímto poděkoval mému vedoucímu diplomové práce Ing. Zdeňkovi Macháčkovi, Ph.D. za rady a čas, který mi poskytl k vypracování této diplomové práce.

Abstrakt

Tato diplomová práce se zabývá vytvořením detekčního systému pro identifikaci osob z obrazového signálu analýzy prstu. Otisky prstu, jež jsou použity v této práci, jsou nasnímány pomocí kontaktního optického snímače, přičemž jako snímač je použita webová kamera.

V řešení jsou použity metody na bázi detekce a extrakce singulárních bodů, které následně slouží k porovnávání otisků prstů. Jednotlivé algoritmy jsou zpracovány v prostředí MATLAB a následně v uživatelském rozhraní GUI se porovnávají nasnímané otisky prstu – vzorový a porovnávaný.

Výsledkem uživatelského rozhraní je informační dialogové okno, jež informuje, zdali je vzorový a porovnávaný otisk prstu totožný. Dále uživatelské rozhraní umožňuje zobrazení výstupů jednotlivých metod.

Klíčová slova

Identifikace osob, otisky prstu, MATLAB, uživatelské rozhraní GUI, singulární body, webová kamera, kontaktní optický snímač.

Abstrakt

This thesis deals with the creation of a detection system for identifying people by image recognition of finger analysis. Fingerprints, which are used in this paper are scanned using contact of optical sensor while as a sensor is used webcam.

In the solutions are used methods based on detection and extraction of singular points, which serves to compare fingerprints. Individual algorithms are processed in MATLAB and then in the user interface of GUI are compared the scanned fingerprints - sample and comparison.

The result of the user interface is an information dialog box which informs whether sample fingerprint and comparison fingerprint are identical. Furthermore, the user interface can display outputs of each method.

Keywords

Identification of person, fingerprints, MATLAB, user interface GUI, the singular points, webcam, contact optical sensor.

Seznam použitých symbolů a zkratek

GUI – grafické uživatelské rozhraní

FRR (False Rejection Rate) - míra chybného odmítnutí

FAR (False Acceptance Rate) - míra chybného přijetí

$G_x(x, y)$ – gradient horizontálního směru

$G_y(x, y)$ – gradient vertikálního směru

$\theta(x, y)$ - lokální orientace obrazu

$\theta'(x, y)$ - lokální orientace obrazu po provedení filtrace dolní propustí

$\phi_x(x, y)$ – komponent vektorové pole pro horizontální směr

$\phi_y(x, y)$ – komponent vektorové pole pro vertikální směr

$\phi'_x(x, y)$ – komponent vektorové pole pro horizontální směr po filtraci dolní propustí

$\phi'_y(x, y)$ – komponent vektorové pole pro vertikální směr po filtraci dolní propustí

$g(x, y)$ – výsledná rovnice Fourierovy transformace

$F(x, y)$ – Fourierova transformace

$f(x, y)$ – inverzní Fourierova transformace

W – velikost bloku $W \times W$

$Core(i, j)$ – maska okolí pro singulární bod “jednoduché” jádro

$DoubleCore(i, j)$ – maska okolí pro singulární bod “dvojitě” jádro

$Delta(i, j)$ – maska okolí pro singulární bod delta

A – vzdálenost mezi singulárním bodem jádro a deltou

B – vzdálenost osy y mezi singulárním bodem jádro a deltou

α – úhel mezi vzdálenostmi A a B

E – Euklidovská vzdálenost

k – korelační koeficient

Obsah

Úvod	1
1. Rozbor problematiky metod identifikace obrazového signálu analýzou prstu	2
1.1 Biometrie otisku prstu	2
1.2 Historie otisku prstu	3
1.3 Analýza otisku prstu	4
1.3.1 Papilární linie	4
1.3.2 Singulární body	5
1.4 Snímače otisku prstu	6
1.4.1 Kontaktní snímače	8
1.4.2 Bezkontaktní snímače	12
1.5 Chyby biometrických systémů	14
1.6 Cena snímačů otisků prstů	15
2. Návrh systému pro identifikace obrazového signálu analýzou prstu	16
2.1 Metody zvýšení kvality obrazu	16
2.1.1 Ekvalizace histogramu	16
2.1.2 Zvýšení kvality obrazu pomocí Fourierovy transformace	17
2.2 Extrakce singulárních bodů jádra a delty	18
2.2.1 Odhad lokální orientace papilárních linií	18
2.2.2 Směrové pole papilárních linií	20
2.3 Porovnání otisků prstů	22
3. Realizace softwaru pro identifikaci obrazového signálu prstu	23
3.1 Matlab	23
3.2 Použité příkazy	23
3.3 Diagram aktivit softwaru pro identifikaci otisků prstů	25
3.4 Načtení obrazu	26
3.5 Předzpracování obrazu	26
3.6 Zvýšení kvality obrazu pomocí Fourierovy transformace	27
3.7 Lokální orientace papilárních linií	28
3.8 Detekce a extrakce singulárních bodů	30

3.9	Porovnání otisků prstů pomocí singulárních bodů	37
3.10	Zobrazení výsledků	41
3.11	Vytvoření a popis uživatelského rozhraní GUI	43
4.	Vizualizace a srovnání naměřených výsledků s teoretickými předpoklady	44
4.1	Srovnání otisků prstů – pravá smyčka (right loop)	45
4.2	Srovnání otisků prstů – levá smyčka (left loop)	46
4.3	Srovnání otisků prstů – závit (whorl)	47
4.4	Srovnání otisků prstů – klenutý oblouk (tented arch)	48
4.5	Srovnání otisků prstů – chybové hlášení	49
5.	Zhodnocení dosažených výsledků	50
5.1	Verifikace systému pro identifikaci otisků prstů – dobrovolník L. P.	50
5.2	Verifikace systému pro identifikaci otisků prstů – dobrovolník P. L.	51
5.3	Verifikace systému pro identifikaci otisků prstů – dobrovolník L. L.	52
5.4	Verifikace systému pro identifikaci otisků prstů – dobrovolník M. S.	53
5.5	Verifikace systému pro identifikaci otisků prstů – dobrovolník F. S.	54
5.6	Verifikace systému pro identifikaci otisků prstů – dobrovolník L. M.	55
5.7	Verifikace systému pro identifikaci otisků prstů – dobrovolník M. R.	56
5.8	Verifikace systému pro identifikaci otisků prstů – dobrovolník D. G.	57
5.9	Verifikace systému pro identifikaci otisků prstů – dobrovolník D. T.	58
5.10	Verifikace systému pro identifikaci otisků prstů – dobrovolník S. P.	59
5.11	Celkové zhodnocení jednotlivých výsledků verifikace	60
6.	Závěr	61
	Použitá literatura	62
	Přílohy	65

Úvod

Identifikace člověka podle analýzy otisku prstu má mnoho využití, zejména jako bezpečnostní prvek u zařízení, u kterých je nutné prokázat totožnost osoby. V minulosti se jako bezpečnostní prvek uvažoval bezpečnostní klíč, několika místní kód, který obsahoval buď čísla, nebo kombinaci čísel a písmen anebo taktéž průkaz totožnosti. Nevýhodou bylo, pokud oprávněná osoba tento kód (bezpečnostní klíč, průkaz totožnosti) zapomněla, tím pádem tato osoba ztratila platný přístup, ke složce, ke které se chtěla dostat anebo prokázat.

V současnosti se začíná používat identifikace podle otisku prstu nebo jiných biometrických údajů jako standard v bezpečnostní složce. Výhodou této metody identifikace je, že osoba si nese své bezpečnostní heslo s sebou jako biometrický údaj, který má osoba od osoby jiný. To znamená, že není možné, aby dvě osoby měly stejný otisk prstu.

Cílem této diplomové práce je vytvoření detekčního systému s identifikací osob z obrazového signálu analýzu prstu. Tento systém bude vytvořen v prostředí MATLAB. Následně se vytvoří uživatelské rozhraní pomocí grafického rozhraní GUI, ve kterém se bude porovnávat jednotlivé obrazy otisku prstů.

Kapitola 1. se zabývá výhodami a nevýhodami použití otisku prstů v identifikačním systému, jeho historií vzniku (kde, kdy a v jakém smyslu byl použit) a také jeho anatomickými proporcemi, jež definují struktury jednotlivých otisků prstů, na kterých bude celý systém stavět. Dále se podíváme na nejrůznější typy snímačů otisků prstů, u kterých si popíšeme jednotlivé kritéria, jež by měl každý snímač otisků dodržet, a také výhody a nevýhody jednotlivých snímačů otisků prstů.

V kapitole 2. se podíváme na tvorbu návrhu systému pro identifikaci obrazového signálu analýzou prstu, kde si ukážeme zjednodušené blokové schéma přístroje pro rozpoznání otisku prstu. Dále jednotlivé metody a algoritmy, jež se použijí pro zpracování a identifikaci otisku prstu, například: ekvalizaci pomocí histogramu, zvýšení kvality obrazu za pomoci Fourierovy transformace.

Ve 3. kapitole se na základě zjištěných poznatků přejde k realizaci softwaru v programu MATLAB. Dále se uvedou jednotlivé použité příkazy a jejich vysvětlivky. Následně si ukážeme diagram aktivit softwaru pro identifikaci otisků prstů, který vysvětluje sled jednotlivých metod, jež jsou použity v programu. A nakonec se vypíšíou jednotlivé algoritmy, které budou popisovat danou metodu, která byla použita v softwaru pro identifikaci otisků prstů.

V 4. kapitole se ověří a srovná funkčnost programu podle teoretických předpokladů. Vizualizace programu bude v prostředí grafického rozhraní GUI, kde se následně budou srovnávat jednotlivé otisky prstů.

V 5. kapitole se provede verifikace systému a to pomocí 10 dobrovolníků, u nichž se nasnímá vzorový otisk prstu a následně tento vzorový otisk prstu se bude porovnávat s nově nasnímaným otiskem prstu. Jednotlivé dosažené výsledky se nakonec napíšíou do tabulky, která na konci bude obsahovat procentuální shodu.

1. Rozbor problematiky metod identifikace obrazového signálu analýzou prstu

1.1 Biometrie otisku prstu

Metoda snímání a rozpoznání lidských charakteristických údajů za účelem autentizace neboli ověření identity měřeného subjektu, se nazývá biometrie[5]. Výhoda této autentizace je v tom, že si člověk nemusí pamatovat několikamístné kombinace hesel nebo mít pořád u sebe např. přihlašovací kartu. Její největší výhodou však je, že biometrické charakteristické znaky zůstávají během života stejné a není možné je odcizit nebo zapomenout[1].

Hlavní náplní identifikačního systému je snímání biometrických údajů a jejich následné porovnávání se šablonami, které byly sejmuty při registraci biometrických údajů. Soudobé identifikační systémy pracují s různými charakteristickými znaky člověka např. otisk prstu, duhovka oka, sítnice oka, geometrie tváře, geometrie ruky, geometrie prstů, struktura žil na zápěstí, složky lidského hlasu, DNA, dynamika podpisu či chůze atd. [1].

Aby tyto biometrické znaky mohli být využity v praxi, musí splňovat určité požadavky[2] :

- *Univerzálnost* – každá osoba by měla mít tyto biometrické znaky
- *Jednoznačnost* – dvě osoby nemají tytéž biometrické znaky
- *Stálost* – biometrické znaky jsou během života stejné
- *Rychlost*
- *Nulové provozní náklady*

Tabulka 1: Porovnání jednotlivých biometrických metod[2]

Biometrická charakteristika	Univerzálnost	Jednoznačnost	Stálost	Shromážditelnost	Výkon	Přijatel-nost	Bezpeč-nost
Obličej	vysoká	nízká	střední	vysoká	nízký	vysoká	nízká
Otisk prstu	střední	vysoká	vysoká	střední	vysoký	střední	střední
Geometrie ruky	střední	střední	střední	vysoká	střední	střední	střední
Oční duhovka	vysoká	vysoká	vysoká	střední	vysoký	nízká	vysoká
Oční sítnice	vysoká	vysoká	střední	nízká	vysoký	nízká	vysoká
Podpis	nízká	nízká	nízká	vysoká	nízký	vysoká	nízká
Hlas	střední	nízká	nízká	střední	nízký	vysoká	nízká

1.2 Historie otisku prstu

Asi nejstarší doklad o existenci papilárních liniích je možné najít u indiánských kmenů, žijících na území dnešního amerického státu Indiana. Na kamenech, jež byly nalezeny, jsou ryté obrazce, tzv. „*petroglyfy*“, vyobrazující lidskou ruku. Tyto petroglyfy jsou staré přibližně několik tisíc let před naším letopočtem. Význam těchto petroglyfů dosud nebyl s jistotou zjištěn. Dále také existují archeologické důkazy, že už minimálně před 6 až 7 tisíci lety před naším letopočtem, byly otisky prstů používány pro identifikační účely a to v Asýrii a Číně. Dále v Babylonu, za královny Hanimurabi (1792 – 1750 před naším letopočtem), byly rovněž k identifikačním účelům použity otisky prstů[4] .

První spis o využití otisků prstů jako prostředku k nalezení totožnosti osob pochází z Číny a jeho autorem byl Číňan Kio Kung-yen. Podle tohoto autora prý Číňané znali účel otisků prstů a používali jej při obchodních transakcích. První použití otisků prstů v kriminálních procesech se zmiňuje v literatuře pocházející z doby Sung (1107 n.l.)[4] .

Na evropském kontinentě byly první zdokumentované vědecké poznatky o otiscích prstů v roce 1686. V tomto roce profesor anatomie, Marcello Malpighi, upozoroval při své práci jednotlivé smyčky, prohlubně a spirály v kůži na konečcích prstů. Poté byla tato vrstva kůže po něm pojmenována[4] .

Asi největší průlom v daktyloskopii (nauka o papilárních liniích [3]) byl v roce 1823, kdy se profesorovi Janu Evangelistovi Purkyně podařilo popsat a rozlišit dvě základní vzory papilárních linií na konečcích prstů[4] .

Mezi těchto devět základních vzorů papilárních linií patří:

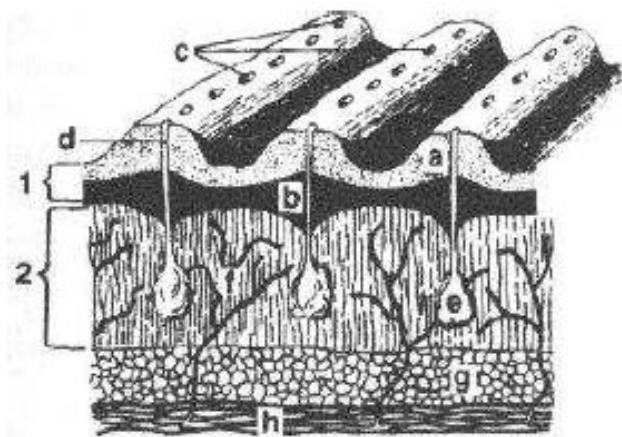
- Příčné záhyby
- Střední podélný pruh
- Šikmý pruh
- Šikmý záliv
- Mandle
- Spirála
- Elipsa
- Kruh
- Zdvojený vrcholek

Daktyloskopie byla poprvé použita v praxi v roce 1903 v newyorské státní věznici. O dva roky později v roce 1905 začala daktyloskopii využívat americká armáda pro identifikaci osob. O další dva roky později námořnictvo[4] .

1.3 Analýza otisku prstu

1.3.1 Papilární linie

Papilární linie jsou vyvýšené povrchové části pokožky každého prstu. Nacházejí se na povrchu dlaní, dlaňové strany prstů a chodidel. Dosahují výšky 0,1 – 0,4 mm a šířky 0,2 – 0,7 mm. Informace o struktuře papilárních linií je zakořeněná v nejhlubší části pokožky a to v zárodečné vrstvě kůže. Díky tomu nejde tato informace lehce odstranit. Byly názory, že seříznutím, spálením či sedřením povrchové vrstvy kůže je možné natrvalo tyto papilární linie odstranit. Dnes je známo, že tomu tak není. Pokud se odstraní nebo poškodí jenom vrchní vrstva kůže, odstraní se papilární linie jen na nějakou dobu, tj. jen do té doby, než se zraněná kůže znovu obnoví. Aby se odstranili papilární linie, musela by se odstranit nejhlubší vrstva kůže, což je zárodečná vrstva[4].



Obr. 1: Schéma stavby kůže s papilárními liniemi[6]

1 – pokožka (epidermis), 2 – škára, a – rohová vrstva, b – zárodečná vrstva, c – póry, d – vývody potních žláz, e – potní žlázy, f – cévy, g – podkožní tuk, h – svalstvo

Na obrázku č. 1 je možné vidět dvě základní vrstvy, vrchní vrstvu epidermis a pod ní vazivovou kůži (škáru). Mezi vrchní a spodní vrstvou se nachází rohová vrstva. Pomezí mezi vrstvami není rovné a tvoří hřebenové útesy a brázdy. Výška hřebenových útesů závisí na velikosti a jemnosti jakou má mít pohyblivost kůže v určitém místě.

Stav a vývoj kůže se rozvíjí již v embryonálním životě. Jednotlivé papilární linie se ukazují mezi čtvrtým a pátým měsícem embryonálního stavu. V této době se tvoří papilární linie, které svůj tvar už nemění po celý život[4].

1.3.2 Singulární body

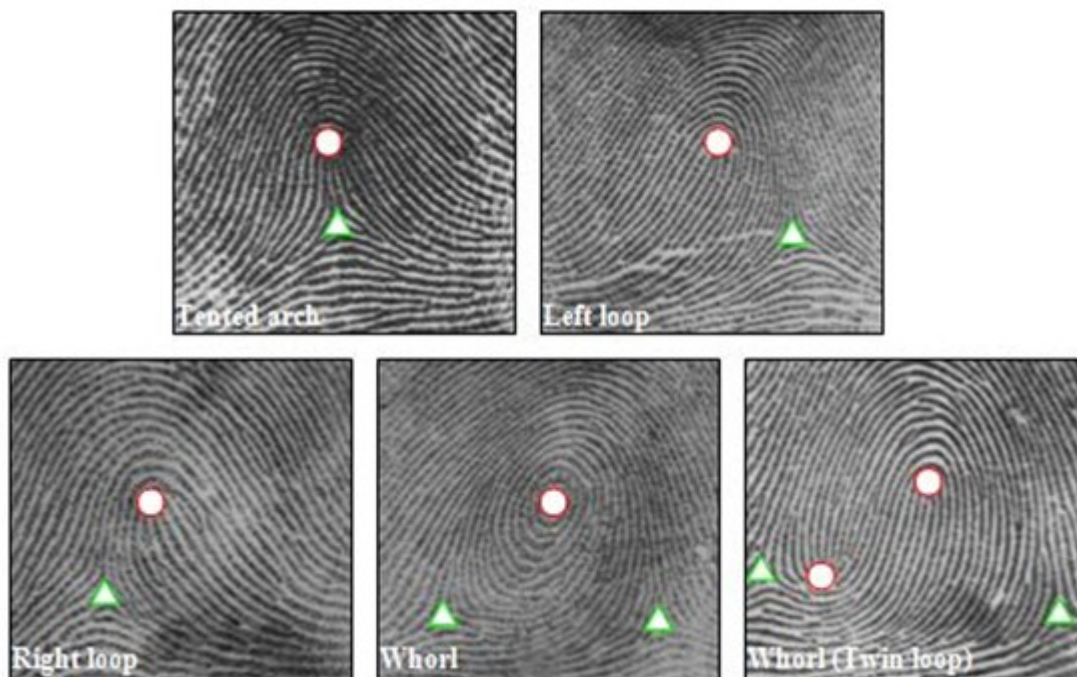
Jak už bylo psáno výše, papilární body tvoří svým rozložením obrazce, jež se třídí do jednotlivých kategorií.

- Závít (whorl)
- Levá smyčka (left loop)
- Pravá smyčka (right loop)
- Dvojitá smyčka (twin loop)
- Klenutý oblouk (tented arch)

Každý z těchto vypsanych kategorií tvoří tzv. singulární body. Tyto singulární body vyznačují zvětšenou přítomností zakřivených papilárních linií a také zvýšenému zhuštění papilárních linií v malém prostoru. Sestava singulárních bodů vytvoří dva typické znaky – bod jádro a delta. V bodě jádro dochází k náhlému otočení směru papilárních linií a to buď o 180° (v případě smyčky) anebo ukončení v uzavřené elipsoidní smyčce (v případě závitu). Naopak bod delta se vyznačuje tím, že ho obklopují papilární linie ze tří různých směrů, které tvoří trojúhelník.

Kombinace těchto singulárních bodů zapříčiní vytvoření výše uvedených kategorií. Tyto kategorie se vyznačují různým počtem bodů delta a bodů jádro. Například otisk typu **klenutý oblouk** se označuje jedním singulárním bodem **delta** a “**jednoduché**” **jádro**. Taktéž jeden singulární bod **delta** a “**jednoduché**” **jádro** obsahuje **levá** a **pravá smyčka**, s tím že rozdíl mezi nimi je v tom, že **levá smyčka** se točí doleva a má na pravé straně singulární bod **delta**, u **pravé smyčky** je to naopak. Dále je **dvojitá smyčka**, která obsahuje dva singulární body **delta** a **jádra**. Nakonec je otisky typu **závit**, který má singulární bod “**dvojité**” **jádro** a dva singulární body **delta**, které jsou symetricky umístěny k singulárnímu bodu “**dvojité**” **jádro**[6].

V navrhovaném algoritmu v této diplomové práci je smyčka **závitu** rozdělena na horní smyčku a spodní, což zapříčiní tomu, že smyčka obsahuje dva singulární body. Tyto body jsou označeny jako jeden singulární bod “**dvojité**” **jádro**.



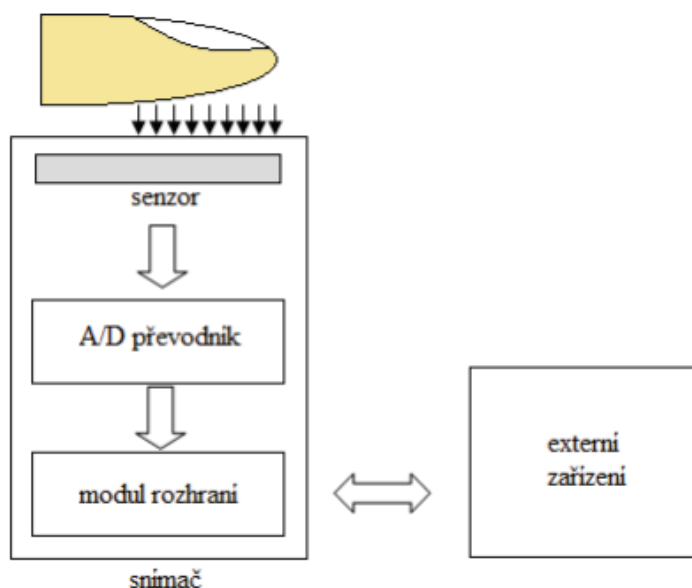
Obr. 2: Zobrazení jednotlivých typů singulárních bodů (jádro a delta)[6].

1.4 Snímače otisku prstu

Zpracování otisků se v minulosti provádělo pomocí specialisty, který analyzoval otisky prstů, jež byly získané pomocí nanesení inkoustu na prst, a poté tento prst byl otisknut na papír. Později s inovací metod zpracování otisků prstů, vznikly metody, jež zahrnovaly výpočetní algoritmy, u kterých je nutné převést jednotlivé otisky prstů do digitální podoby. Aby se obraz převedl do digitální podoby, lze použít dvě metody[6].

První metoda je za pomoci scanneru, kdy se naskenují jednotlivé otisky prstů získané z inkoustového obrazu. Tato metoda je stále používána v kriminalistické praxi při konvertování klasických získaných otisků prstů, za pomoci inkoustu, do databáze. Druhá metoda je za pomoci elektronických snímačů, které jednotlivé otisky nasnímají přímo z prstu a uchovají je v digitální podobě. Obě dvě ty metody se nazývají offline snímání[6].

V dnešní době je mnohem používanější metoda druhá. Stavba běžného elektronického snímače se skládá z několika bloků, které jsou zobrazeny na obr. 5. Senzor snímá složení papilárních linií z přitisknuté plochy prstu a za pomoci A/D převodníku převádí analogový signál ze senzoru do digitální podoby. Modul rozhraní obstarává realizaci komunikace mezi externím zařízením a snímačem. Jako externí zařízení může sloužit například osobní počítač[7].



Obr. 3: Schéma blokového diagramu běžného typu snímače otisku prstu[7]

Elektronický snímač může být popsán podle několik kritérií[8] :

- **Plocha senzoru** – dostatečná velká plocha pro snímání. Je to z důvodu zachycení co největší plochy otisku prstu.
- **Rozměr zařízení** – možnost využití v mobilních zařízeních.
- **Rozlišení** – počet pixelů na palec, zkratka dpi. Standard je 500 dpi.
– čím větší bude rozlišení snímače, tím bude kvalitnější výstupní obraz.
- **Opakovatelnost** – senzor by měl být navržen, tak aby při každém opětovném snímání došlo k minimálnímu posunu nebo natočení snímaného prstu. To znamená, že konstrukce snímače musí přesně vymežit, jak, kde, pod jakým úhlem a z která strany, má uživatel vložit snímaný prst. A to proto, aby vložený prst ke snímání, byl načten ze stejného místa jako v předchozím snímání.
- **Mechanická a elektrická odolnost** – snímač nesmí být zničen špatnou manipulací či elektrostatickým výbojem.
- **Ochrana proti napodobeninám** – snímač by měl obsahovat systémy, jež hlídají teplotu, tep, kapacitu a jiné fyziologické parametry, zamezující použití napodobeniny k proniknutí do systému.
- **Cena** – cena senzoru by měla být v přijatelných mezích. Neměla by překročit například cenu zařízení, do kterého má být použita.

Elektronický snímač se může dělit podle typu sejmutí otisku prstů[8] :

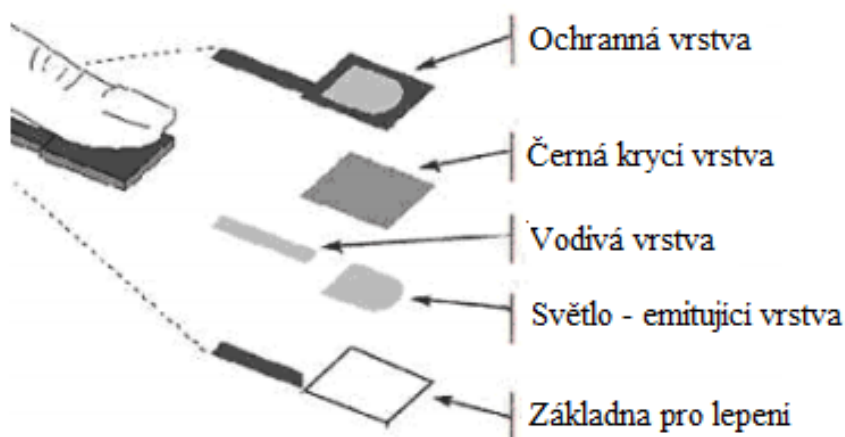
- Kontaktní
- Bezkontaktní

1.4.1 Kontaktní snímače

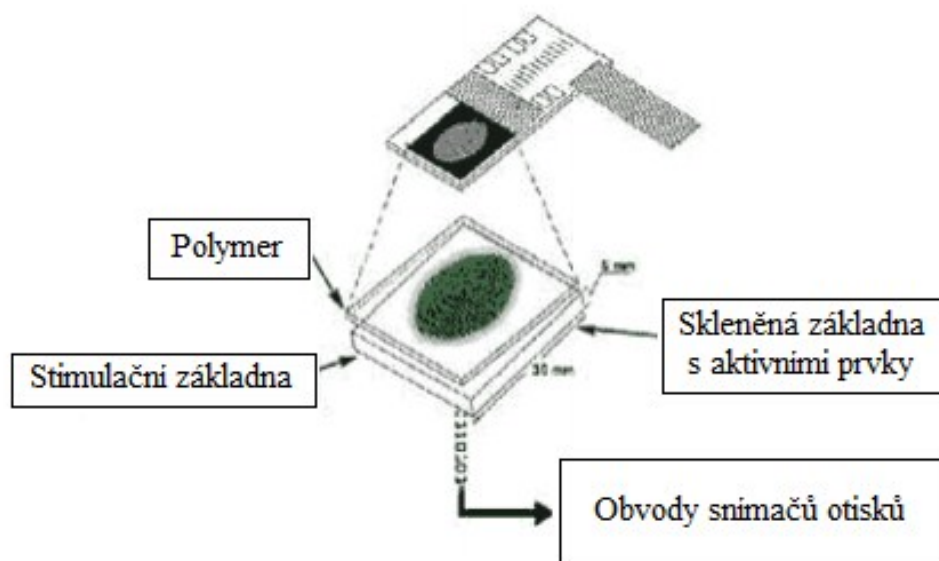
1.4.1.1 Elektroluminiscenční snímač

Elektroluminiscenční snímač patří nejnovější technologii ve snímání otisků prstů. Snímací plocha, na kterou se vkládá prst, používá vrstvu polymeru, která obsahuje několik vrstev. Tato technologie pracuje na základě světlo – emitující vrstvy, která se nachází v polymeru a na změně tlaku přiloženého prstu. Po přiložení prstu na elektroluminiscenční snímač dojde vlivem zvýšeného tlaku papilárních linií k emitování světla na světlo – emitující vrstvě. To znamená, že jednotlivé prohlubně mezi papilárními liniemi se nezobrazí na světlo – emitující vrstvě. Pro převedení obrazu papilárních linií do digitální podoby slouží fotodiody, které detekují světlo a jejich výstupem je elektrický signál[9] .

Výhodou elektroluminiscenčních snímačů jsou malé rozměry, dobré rozlišení a přijatelná cena. Další výhodou je, že i s použitím prstu, který je suchý, kvalita otisku se nijak nesnižuje. Na druhou stranu, jeho nevýhoda je nízká odolnost proti mechanickému poškození a také tendence ke znečištění prachem nebo vodou[9] .



Obr. 4: Jednotlivé vrstvy polymeru tvořící plochu snímače[9] .



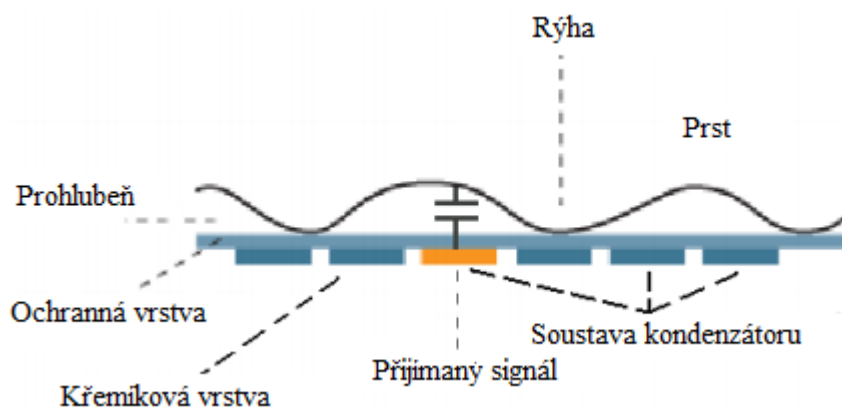
Obr. 5: Konstrukce elektroluminiscenčního snímače[9] .

1.4.1.2 Kapacitní snímač

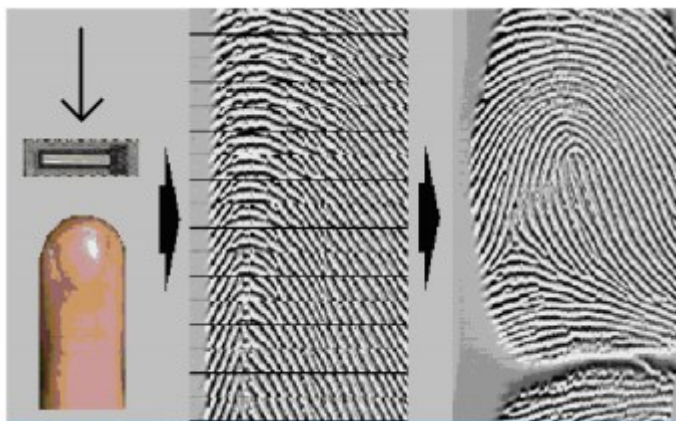
Snímač obsahuje velký počet soustav kondenzátoru, jež jsou od sebe izolovány. Tyto soustavy tvoří jednu část kondenzátoru. Tu druhou část tvoří přiložený prst. Přiložený prst obsahuje papilární linie, které při přiložení na vodivou plochu, vytvoří vodivý kontakt. Tento vodivý kontakt způsobí změnu kapacity jednotlivých kondenzátorů. Naopak prohlubně, které se nedotýkají vodivé plochy, působí jako izolant. Díky tomuto senzor snímá jenom papilární linie, což vlastně je záměr[8] .

Snímače mohou být zkonstruovány ve dvou verzích a to jako plošný snímač anebo ve formě snímače, používajícího tzv. metodu sweepingu. Princip této metody je ve snímání a následném sestavení obrazu z jednotlivých řádků. Tyto jednotlivé řádky se pomocí softwaru složí do výsledného obrazu otisku prstů[8] .

Snímač má několik výhod. Jednou z těchto výhod je vysoká bezpečnost, která tkví v tom, že přes kapacitní snímač se musí přejet prstem. Další výhodou jsou malé rozměry snímače, díky tomu se kapacitní snímač používá v mobilních zařízeních jako je notebook nebo PDA. Nevýhodou tohoto snímače je krátká životnost z důvodu zničení statickou elektřinou[8] .



Obr. 6: Princip kapacitního snímače otisku prstů[8] .

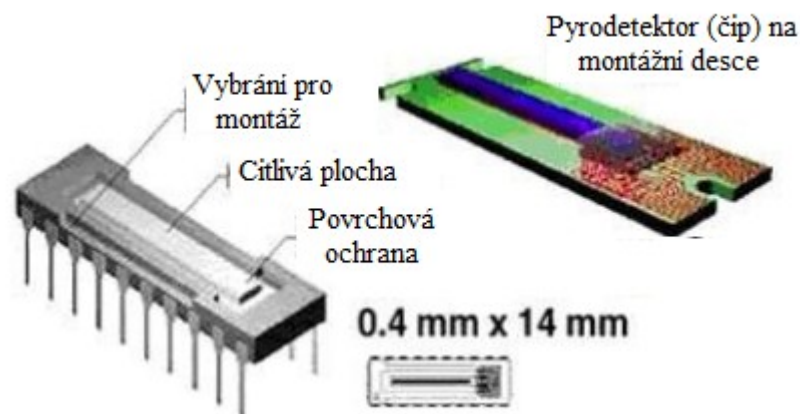


Obr. 7: Postup sestavení obrazu otisku prstu z jednotlivých řádků[8] .

1.4.1.3 Teplotní snímač

Teplotní snímač se skládají z miniaturního, velmi citlivého čipu, jenž se nazývá pyrodetektor. Princip snímání pracuje na rozdílu teplot mezi jednotlivými papilárními liniemi a jednotlivými prohlubněmi, které jsou mezi těmito liniemi. Aby se získal požadovaný otisk prstu, je zapotřebí přejít vloženým prstem přes citlivou plochu. Tato plocha má rozměry 0,4 x 14 mm. Výstupní obraz snímače je formě digitálních pásů. Z těchto pásů se následně pomocí softwaru poskládá požadovaný otisk prstu[9] .

Jelikož snímač snímá teplotu, je nutné, aby vložený prst byl „živý“. Díky tomu má tento snímač dobrou bezpečnost. Na druhou stranu jeho nevýhodou je malá plocha citlivé vrstvy. Jelikož se prst snímá přes takto malou citlivou vrstvou, dochází ke špatnému nasnímání otisku a následnému snímání různých částí prstu, což ve výsledcích může způsobit, že oprávněný uživatel není identifikován a tím pádem je odmítnut. Díky tomuhle se teplotní snímač nepoužívá v přístupových systémech, kde se klade na vysokou bezpečnost. Naopak se používá např. v noteboocích, kde není kladen nárok na tak vysokou bezpečnost[9] .

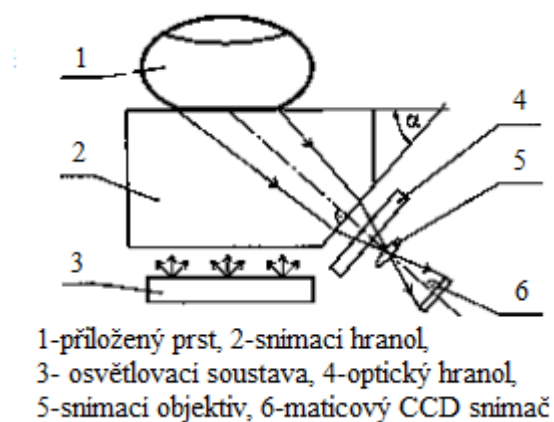


Obr. 8: Obraz teplotního snímače[9] .

1.4.1.4 Optický snímač

Optický snímač patří mezi nejpoužívanější snímače otisku prstu. Princip optického snímače je založen na snímání rozdílného odrazu světla, resp. rozptylu světla na rozhraní snímací plochy hranolu a přiloženého prstu. Odraz světla vznikne pomocí optického hranolu, který se osvětluje pomocí osvětlovací soustavy (viz. Obr. 11) například několika diodami. Světlo vytvořené těmito diodami se pomocí optického hranolu nasměruje na plochu pod přiloženým prstem. Na rozhraní mezi přiloženým prstem a plochou hranolu dojde odrazu světla. Odražené světlo neobsahuje odraz světla od prohlubní, jelikož tento odraz zastínily papilární linie a to znamená, že se na výstupu zobrazí jen papilární linie. Následně odražené světlo míří přes optické filtry do CCD snímače. Vytvořený obraz je následně digitalizován a zpracován algoritmem pro rozpoznávání obrazu[10] .

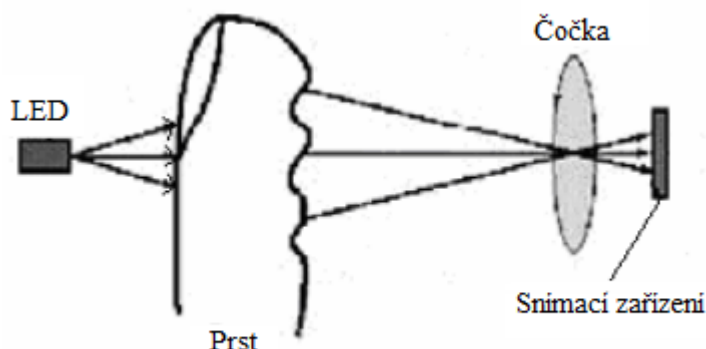
Výhodou tohoto optického snímače je dobré rozlišení a kvalita obrazu. Nevýhodou tohoto snímače však je, že u levných snímačů dochází ke znečištění snímací plochy, jež vede ke zkreslení následujícího snímání otisku budoucího uživatele a následnému zamítnutí. U některých snímačů je přidána speciální silikonové fólie, která zlepší kvalitu suchých otisků prstů[10] .



Obr. 9: Princip optického snímače[10]

1.4.1.5 Transmisní snímač

Princip činnosti transmisního snímače spočívá ve snímání otisku prstu, jenž je prosvětlován z vrchní strany. Zdroj světla je běžně infračervená LED dioda. Snímacím prvkem je CMOS nebo CCD snímač[8] .



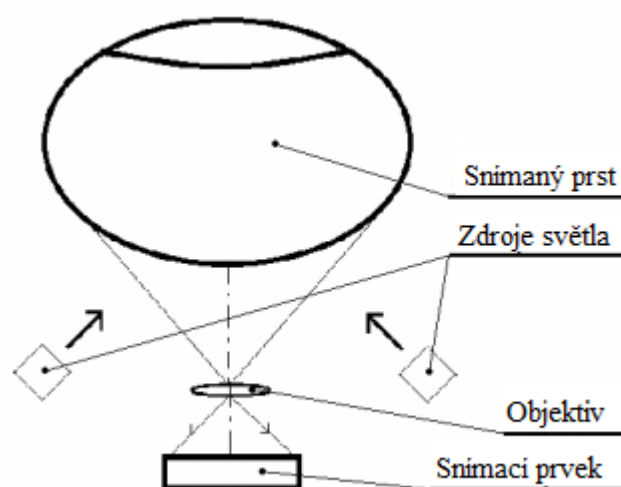
Obr. 10: Princip transmisního snímače[8]

1.4.2 Bezkontaktní snímače

1.4.2.1 Optický snímač

Rozdíl mezi kontaktním a bezkontaktním optickým snímačem je v tom, že bezkontaktní optický snímač nepotřebuje optický hranol. Princip funkčnosti spočívá ve snímání přiloženého prstu, který je v určité vzdálenosti nad CMOS nebo CCD snímačem. Přiložený prst se pod úhlem 45° osvětluje z obou stran diodami, díky toho se papírární linie zviditelní. Následně se osvětlený prst zobrazí na CMOS nebo CCD snímači[1],[8] .

Výhodou tohoto snímače je, že nepotřebuje optický hranol, což následně snižuje pořizovací cenu. Díky tomu, že snímač neobsahuje optický hranol, nedojde ke znečištění snímací plochy. Nevýhoda je v tom, že snímání přiloženého otisku nemusí být ve stejné poloze a vzdálenosti jako u předchozího snímání, což může vést k zamítnutí uživatele. Tato nevýhoda se řeší zavedením stojanu, na kterém se přiloží prst, tím pádem prst je ve stejné poloze a výšce jako u předchozího snímání[1],[8] .

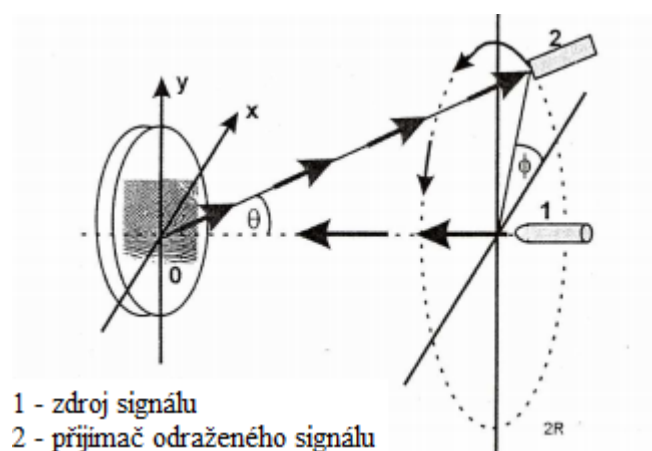


Obr. 11: Princip bezkontaktního optického snímače[8].

1.4.2.1 Ultrazvukový snímač

Princip ultrazvukového snímače je ve vysílání ultrazvukových vln ($4 - 25$ MHz), které jsou generovány ve směru snímaného otisku prstu. Ultrazvukové vlny, které se odrazí od přiloženého prstu, jsou přijímány přijímačem, který je umístěn kolmě k vysílaným vlnám. Odražené ultrazvukové vlny jsou následně vyhodnocovány na základě závislosti mezi dopadajícími a odraženými zvukovými vlnami. Výsledný obraz otisku prstu má vysoký kontrast s přesností $0,1$ mm a navíc je trojrozměrný[11].

Mezi výhody ultrazvukového snímače patří odolnost vůči vlhkým otiskům prstů, znečištění nebo obroušení slabé vrstvy kůže. Jelikož ultrazvukový snímač vytváří trojrozměrný obraz, lze snadno rozlišit napodobeninu otisku prstu, jež je dvojrozměrná[11].



Obr. 12:Princip ultrazvukového snímače[11].

1.5 Chyby biometrických systémů

Cílem biometrických systémů je umožnit přístup uživatelům, u nichž byla ověřena identita oprávněného uživatele a zamítnout neoprávněné osoby[11].

Aby systém spolehlivě pracoval, jsou dané dva parametry, jež popisují chybovost neboli spolehlivost systému[9] ,[11]:

-FRR (False Rejection Rate, míra chybného odmítnutí)

- Jedná se o pravděpodobnost, s kterou systém zamítne přístup oprávněné osobě.

$$FRR = \frac{\text{počet osob, které systém chybně zamítnul}}{\text{celkový počet pokusů autorizovaných osob}} \quad (1.1)$$

-FAR (False Acceptance Rate, míra chybného přijetí)

- Jedná se o pravděpodobnost, s kterou systém povolí přístup neoprávněné osobě.

$$FAR = \frac{\text{počet osob, které systém chybně zamítnul}}{\text{celkový počet pokusů neoprávněných osob}} \quad (1.2)$$

Rozdělení míry spolehlivosti: Vysoká – FAR < 0,01 %

Střední – 0,1 < FAR < 1 %

Nízká – FAR > 1 %

Tabulka 2: Hodnoty jednotlivých spolehlivostí biometrických systémů[11].

Biometrická metoda	FRR	FAR	Rychlost verifikace	Míra spolehlivosti
Otisk prstu	< 1,0%	0,0001% - 0,00001%	0,2 až 1 sekunda	vysoká
Geometrie ruky	< 0,1%	0,1%	1 až 2 sekundy	střední
Tvář	1%	0,1%	3 sekundy	střední
Oční sítnice	< 0,4 %	0,001 %	1,5 až 4 sekundy	vysoká
Oční duhovka	0,00066 %	0,00078 %	2 sekundy	vysoká
Hlas	< 1,0%	> 1,0%	1,5 sekundy	nízká

1.6 Cena snímačů otisků prstů

Cena snímačů otisků prstů závisí hlavně na typu snímače (například jestli je to optický nebo kapacitní snímač), v jakých podmínkách má pracovat, na rychlosti identifikace nebo také z jakého materiálu je tento snímač vyroben. V tabulce č. 3 jsou pro ukázkou znázorněny pár parametrů snímačů, jako je typ snímače, rozlišení, operační systém, operační teplota a nakonec jeho cena.

Tabulka 3: Hodnoty parametrů jednotlivých biometrických systémů [12]

Název	Typ snímače	Rozlišení (dpi)	OS	Oper. Teplota	Cena (Kč)
Lumidigm Mercury M30x	Optický	500	Windows, Linux	(-10)°C až +60°C	5 477
Fingkey Hamster II	Optický	500	Windows, Linux	(-10)°C až +55°C	2 213
Verifi P5100	Kapacitní	508	Windows	0°C až + 70°C	3 331
id3 Certis Image	Teplotní	500	Windows	0°C až + 50°C	12 855
ASEDrive IIIe Combo Bio F2	Kapacitní	500	Windows		4 270

Lumidigm Mercury M30x

- Úložná kapacita až 50 000 uživatelů
- Obsahuje multispektrální technologii, díky níž pracuje i zhmožděné kůži

Verifi P5100

- Hliníkové pouzdro odolné vůči poškrábání

Fingkey Hamster II

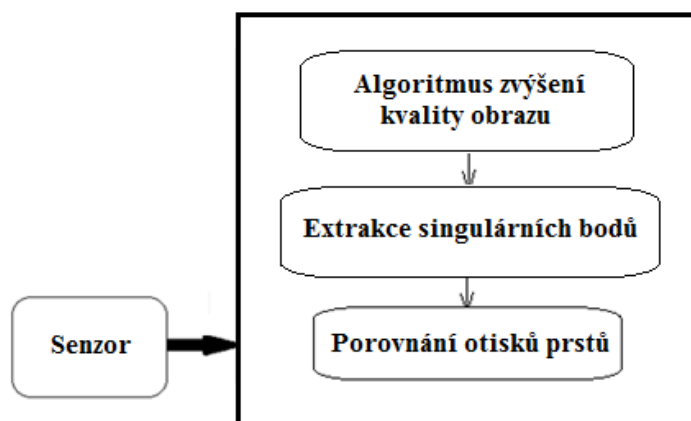
- Zabraňuje použití falešných otisků prstů
- První místo v soutěži FVC (Fingerprint Verification Competition)
- Vysoká účinnost u mokrých a suchých prstů

Id3 Certis Image

- Imunitní vůči elektrostatickým výbojům
- Skoro není možnost ošidit systém pomocí umělého prstu

2. Návrh systému pro identifikace obrazového signálu analýzou prstu

Systém pro rozpoznání otisků prstů obsahuje snímač otisků prstů (pomocí něhož se nasnímají otisky prstů), dále algoritmy pro zvýšení kvality obrazu, extrakci jádra a delty, a nakonec porovnání jednotlivých otisků prstů (viz. Obr. 13).



Obr. 13: Zjednodušené blokové schéma přístroje pro rozpoznání otisku prstu

- Snímače, jež se používají v systémech pro rozpoznání otisků prstů, byly popsány výše.

2.1 Metody zvýšení kvality obrazu

Jelikož nasnímaný obraz otisku prstu není vždy v nejlepší kvalitě, je nutné provést různé změny obrazu, které slouží k zvýšení kvality a následnému zvýraznění jednotlivých složek otisku prstu.

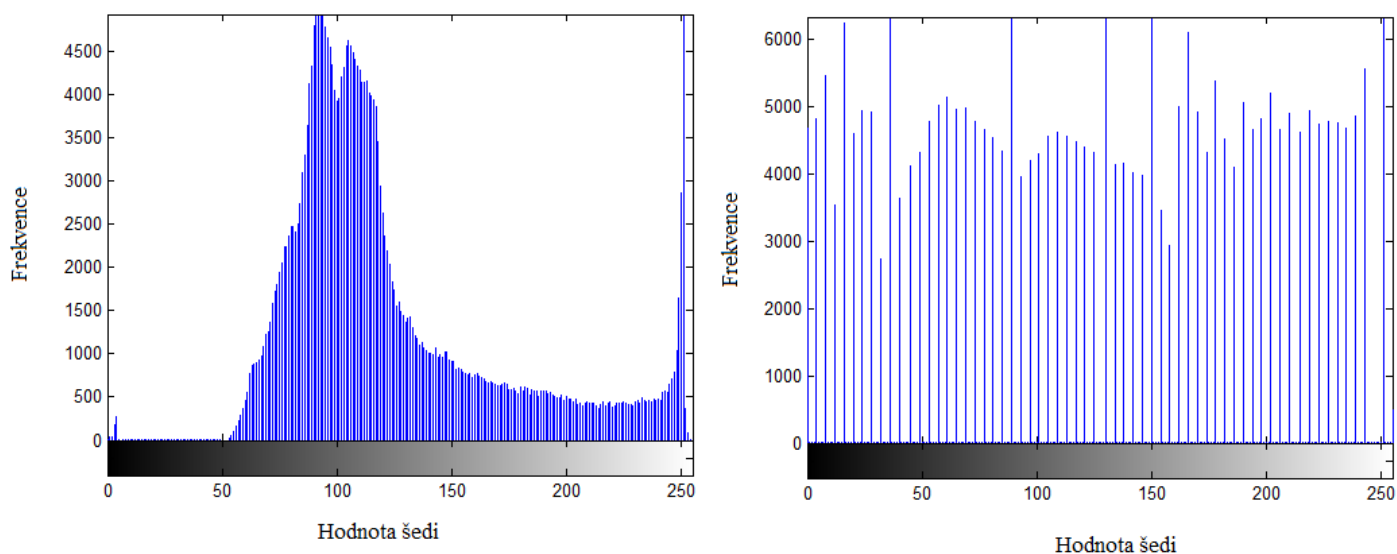
Metody, jež jsou použity v této práci pro zvýšení kvality obrazu, jsou dvě:

- Ekvalizace histogramu
- Fourierova transformace

2.1.1 Ekvalizace histogramu

Slouží k rozšíření distribuce hodnot pixelů obrazu, tak aby se zvýšila percepční informace, tzv. upravuje kontrast obrazu s použitím jeho histogramu[13] [14] . To znamená, že zvyšuje kontrast pro úrovně jasu, jež jsou blízko maxim histogramu, a snižuje kontrast blízko minim histogramu[16] .

Na obr. 14 je znázorněna ukázka histogramu (na pravé straně) a jeho ekvalizace (na levé straně).



Obr. 14: Ukázka před a po ekvalizaci histogramu

2.1.2 Zvýšení kvality obrazu pomocí Fourierovy transformace

Zvýšení kvality obrazu se provádí pomocí Fourierovy transformace a to tak, že se obraz rozdělí do jednotlivých bloků ($W \times W$) a následně se udělá FFT jednotlivých bloků za pomoci rovnice (2.1) – (2.3) [13] .

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \cdot \exp \left\{ -j2\pi \cdot \left(\frac{ux}{M} + \frac{vy}{N} \right) \right\} \quad (2.1)$$

, kde $u, v = 0, 1, 2, \dots, W$.

Následně se FFT jednotlivých bloku vynásobí velikosti originálního FFT

$$g(x, y) = F^{-1} \{ F(u, v) \cdot |F(u, v)|^K \} \quad (2.2)$$

, kde $F^{-1} F(u, v)$

$$f(x, y) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} F(u, v) \cdot \exp \left\{ -j2\pi \cdot \left(\frac{ux}{M} + \frac{vy}{N} \right) \right\} \quad (2.3)$$

, pro $x, y = 0, 1, 2, \dots, W$.

K v rovnici (2.2) znázorňuje experimentálně určenou konstantu, která byla zvolena $K = 0.45$. Vyšší hodnota konstanty K zlepší kvalitu papilárních linií, ale na druhou stranu, když konstanta K bude příliš vysoká, může dojít ke spojení papilárních linií [13] . V tomto případě

slouží rovnice (2.1) – (2.3) jako dolní propust, která vyfiltruje vysokofrekvenční složku, což způsobí, že se obraz vyhladí[17] .

2.2 Extrakce singulárních bodů jádra a delty

Extrakce singulárních bodů je důležitým krokem pro klasifikaci a následného porovnání otisků prstů. V následujících podkapitolách se vypíší jednotlivé metody, díky kterým se dosáhne k detekci a následné extrakci singulárních bodů.

2.2.1 Odhad lokální orientace papilárních linií

Odhad lokální orientace slouží ke stanovení směru papilárních linií. Tato lokální orientace je nutná k detekci a následné extrakci singulárních bodů.

Popis algoritmu lokální orientace papilárních linií[18] :

- a) Rozdělení obrazu Image (x, y) na bloky o velikosti w x w.
- b) Výpočet gradientů $G_x(x, y)$ a $G_y(x, y)$ za pomoci Sobelovy konvoluční masky[22] .

-1	0	1
-2	0	2
-1	0	1

-1	-2	-1
0	0	0
1	2	1

Obr. 15: Sobelova konvoluční maska pro horizontální a vertikální směr

- c) Odhad lokální orientace obrazu Image (x, y):

$$G_{xx} = \sum_{(x,y) \in w} G_x^2(x, y) \quad (2.4)$$

$$G_{yy} = \sum_{(x,y) \in w} G_y^2(x, y) \quad (2.5)$$

$$G_{xy} = \sum_{(x,y) \in w} G_x(x, y) \cdot G_y(x, y) \quad (2.6)$$

$$\theta(x, y) = \frac{1}{2} \tan^{-1} \left(\frac{2G_{xy}}{G_{xx} - G_{yy}} \right) \quad (2.7)$$

- d)** Následně kvůli šumu, přerušených hřebenů papilárních linií a dalším poškozením v obraze, se provede filtrace filtrem typu dolní propust. Proto, aby se provedla filtrace typu dolní propust, musí se obraz převést do spojitého vektorového pole:

$$\phi_x(x, y) = \cos(2\theta(x, y)) \quad (2.8)$$

$$\phi_y(x, y) = \sin(2\theta(x, y)) \quad (2.9)$$

, kde ϕ_x a ϕ_y jsou x a y komponenty vektorového pole.

$$\phi'_x(x, y) = \sum_{u=-\frac{W_\phi}{2}}^{\frac{W_\phi}{2}} \sum_{v=-\frac{W_\phi}{2}}^{\frac{W_\phi}{2}} W_\phi(u, v) \cdot \phi_x(x - u, y - v) \quad (2.10)$$

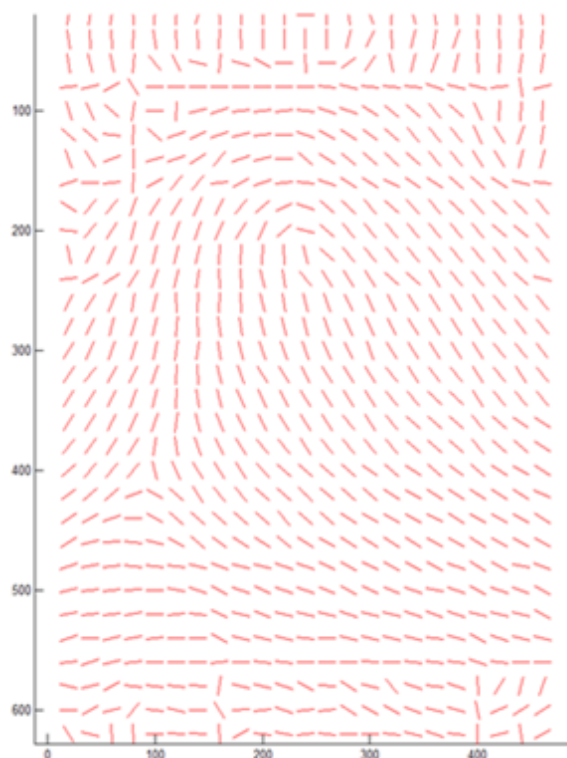
$$\phi'_y(x, y) = \sum_{u=-\frac{W_\phi}{2}}^{\frac{W_\phi}{2}} \sum_{v=-\frac{W_\phi}{2}}^{\frac{W_\phi}{2}} W_\phi(u, v) \cdot \phi_y(x - u, y - v) \quad (2.11)$$

- e)** Lokální orientace hřebenů papilárních linií v bodě (x, y):

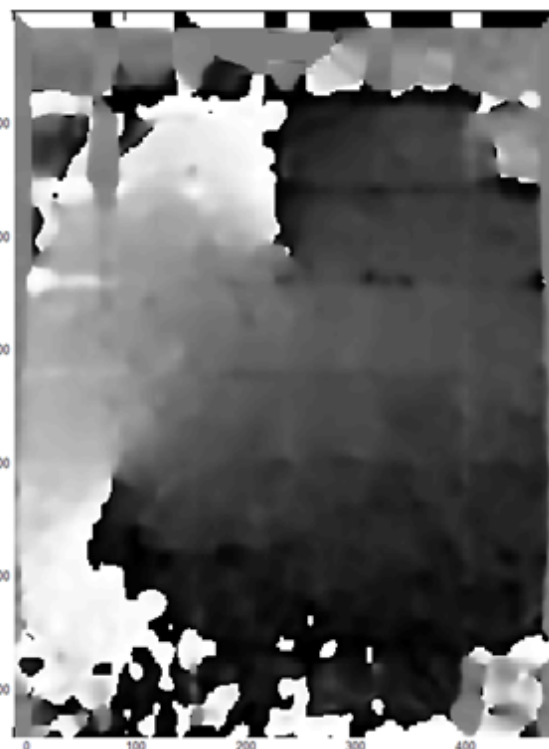
$$\theta'(x, y) = \frac{1}{2} \tan^{-1} \left(\frac{\phi'_y(x, y)}{\phi'_x(x, y)} \right) \quad (2.12)$$

2.2.2 Směrové pole papilárních linií

Po provedené lokální orientaci obrazu se vytvoří směrové pole, které představuje aktuální směr papilárních linií, viz níže.



Obr. 16: Orientace papilárních linií



Obr. 17: Směrové pole papilárních linií

Na obr. 16 je znázorněna orientace papilárních linií otisku prstu, po provedení výpočtů rovnic (2.4) – (2.7). Na obr. 17 je zase znázorněno směrové pole, ke kterému se došlo pomocí rovnic (2.8) – (2.12).

Popis algoritmu detekce a extrakce bodů jádro:

- Převod hodnot pixelů (obr. 17), jež jsou v radiánech na stupně ($0 - 180^\circ$).
- Pomocí masky okolí se v obr. 17 bude hledat singulární bod jádro, za pomoci rovnice:

$$Core(i, j) = P_k < P_{k+1} \quad (2.13)$$

, kde $k = 1, \dots, 7$

P7	P8	P1
P6	i, j	P2
P5	P4	P3

Obr. 18: Maska okolí pro určení singulárního bodu – jádro (horní smyčka)

- c) V případě, že se jedná o otisk typu závit (whorl) nebo dvojité smyčka (twin loop), hledá se jádro i pro dolní smyčku a to pomocí rovnice:

$$DoubleCore(i, j) = P_k < P_k + 1 \quad (2.14)$$

, kde $k = 1, \dots, 7$

P3	P4	P5
P2	i, j	P6
P1	P8	P7

Obr. 19: Maska okolí pro určení singulárního bodu – jádro (dolní smyčka)

- d) Nalezení bodu delta za pomoci rovnice:

$$Delta(i, j) = P_k > P_k + 1 \quad (2.15)$$

, kde $k = 1, \dots, 7$

P3	P4	P5
P2	i, j	P6
P1	P8	P7

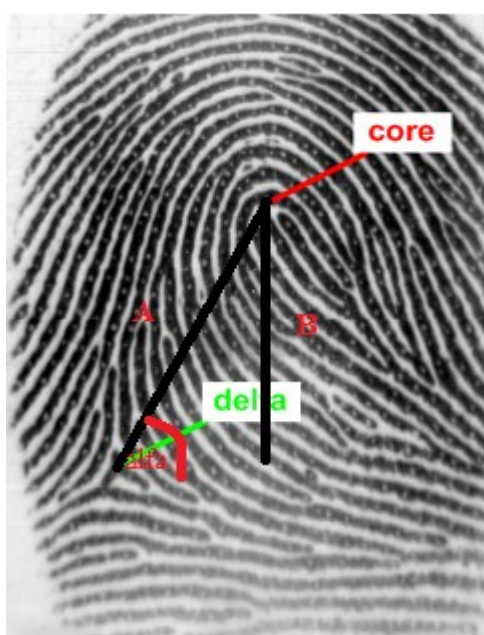
Obr. 20: Maska okolí pro určení singulárního bodu - delta

2.3 Porovnání otisků prstů

Porovnání otisků prstů funguje na principu porovnání pozice singulárních bodů, úhlů a kategorií otisku prstu, viz.: níže.

Popis algoritmu porovnání otisků prstů:

- Nejprve se porovná, jestli oba dva otisky patří do stejné kategorie, například (závit – závit).
- Dále se vypočte vzdálenost A a B. Z těchto vzdáleností se dále vypočte úhel alfa.
- Pokud rozdíl vzdálenost A vzorového a porovnávaného otisku prstu bude menší, než 5 pixelů, rozdíl úhlu nebude větší, než 8° a oba dva otisky patří do stejné kategorie – jedná se o dva stejné otisky prstů.



Obr. 21: Ukázka metody výpočtů porovnání otisků prstů

- Pokud rozdíl vzdálenost A vzorového a porovnávaného otisku prstu bude větší, než 5 pixelů a menší, než 20 pixelů ($5 < A < 20$), spočte se korelační koeficient směřového pole. Pokud tento korelační koeficient bude mít víc, jak 40,55 %, rozdíl úhlu nebude větší, než 8° a oba dva otisky patří do stejné kategorie – jedná se o dva stejné otisky prstů.

Vzdálenost se vypočte pomocí rovnice Euklidovské vzdálenosti:

$$E = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2} \quad (2.16)$$

Korelační koeficient se vypočte pomocí rovnice:

$$k = \frac{\sum_m \sum_n (A_{mn} - \bar{A}) \cdot (B_{mn} - \bar{B})}{\sqrt{(\sum_m \sum_n (A_{mn} - \bar{A}))^2 \cdot (\sum_m \sum_n (B_{mn} - \bar{B}))^2}} \quad (2.17)$$

, kde \bar{A} – střední hodnota A, \bar{B} – střední hodnota B

3. Realizace softwaru pro identifikaci obrazového signálu prstu

V této kapitole bude probána realizace jednotlivých metod a algoritmů, které byly popsány v předchozí kapitole. Jednotlivé metody a algoritmy budou zpracovány pomocí programu MATLAB, ve kterém bude napsán kód, který bude obsahovat příkazy, jež jsou popsány níže, viz.: 3.2 Použité příkazy. Dále bude tato kapitola obsahovat diagram aktivit, jež bude popisovat sled funkce programu. Pomocí tohoto diagramu aktivit budou popsány jednotlivé metody, podle kterých bude pracovat systém pro identifikaci otisků prstů.

3.1 Matlab

MATLAB ® (*matrix laboratory*) je programovací jazyk a interaktivní prostředí pro numerické výpočty, vizualizaci a programování. Použitím programu MATLAB, je umožněno analyzovat data, vývoj algoritmů a vytváření modelů a aplikací. Jazyk, nástroje a vestavěné matematické funkce umožní prozkoumat více přístupů a dospět k řešení rychleji, než s tabulkami nebo tradiční programovací jazyky. MATLAB umožňuje používat celou řadu aplikací, včetně zpracování signálu a komunikací, zpracování obrazu a videa, řídicích systémů, testování a měření, výpočetní finančnictví a výpočetní biologie[20] . Hlavní oblastí, kde je tento program využit, jsou technické obory a ekonomie[19] .

3.2 Použité příkazy

Uigetfile - slouží pro otevření standartních dialogových oken pro získání určitých souborů. Přesněji zobrazí modální dialogové okno se seznamem souborů v aktuální složce, a umožňuje nastavit nebo zadat název souboru. Je-li název souboru platný, a pokud soubor existuje, vrátí *uigetfile* název souboru jako řetězec při klepnutí na tlačítko **Otevřít**. V opačném případě *uigetfile* zobrazí příslušnou chybovou zprávu, po kterém řízení se vrací do dialogového okna.

Imread – slouží k načtení grafického souboru.

Histeq – slouží ke zvýšení intensity kontrastu obrazu.

Size – vrátí numerickou hodnotu, který vypovídá o počtu řádků a sloupců ve zkoumané matici.

Double – převedení vstupní proměnné do datového typu o přesnosti double.

Zeros – vytvoření matice, jež obsahuje nulové hodnoty.

FFT2 – 2- D (dvoudimenzionální) rychlá Fourierova transformace.

IFFT2 - 2- D (dvoudimenzionální) inverzní rychlá Fourierova transformace.

Max – tento příkaz vrátí maximální hodnotu, jež obsahuje vstupní proměnná.

Abs – tento příkaz vrátí absolutní hodnotu vstupní proměnné.

Rgb2gray - příkaz slouží k převedení souboru ve formátu RGB do formátu odstínu šedi (šedotónové stupnice).

Mod – tento příkaz vrátí zůstatek po dělení.

Filter2 – tento příkaz vytvoří 2-D (dvoudimenzionální) FIR filtr.

Fspecial - tento příkaz vytvoří předdefinovaný 2-D (dvoudimenzionální) filtr.

Imfilter - tento příkaz N-D filtruje mnohorozměrné obrazy.

Transpose – transponování vstupní matice.

Atan2 - čtyř-kvadrantový arkustangens

Find – slouží pro nalezení indexů a hodnot nenulových prvků.

Length – tento příkaz vrátí numerickou hodnotu délky dané vstupní proměnné.

Sort – tento příkaz slouží k seřazení prvků vstupní matice.

Bwlabel – tento příkaz slouží k označení spojených komponentů v 2-D (dvoudimenzionálním) binárním obraze

Median – tento příkaz vrátí numerickou hodnotu mediánu vstupní proměnné.

Round – tento příkaz slouží k zaokrouhlení numerické hodnoty vstupní proměnné.

Isempty – tento příkaz zjistí, zdali vstupní proměnná je prázdná či ne. Jestliže je vstupní proměnná prázdná, výstupem bude logická 1, v opačném případě logická 0.

Sqrt – tento příkaz odmocní vstupní proměnnou.

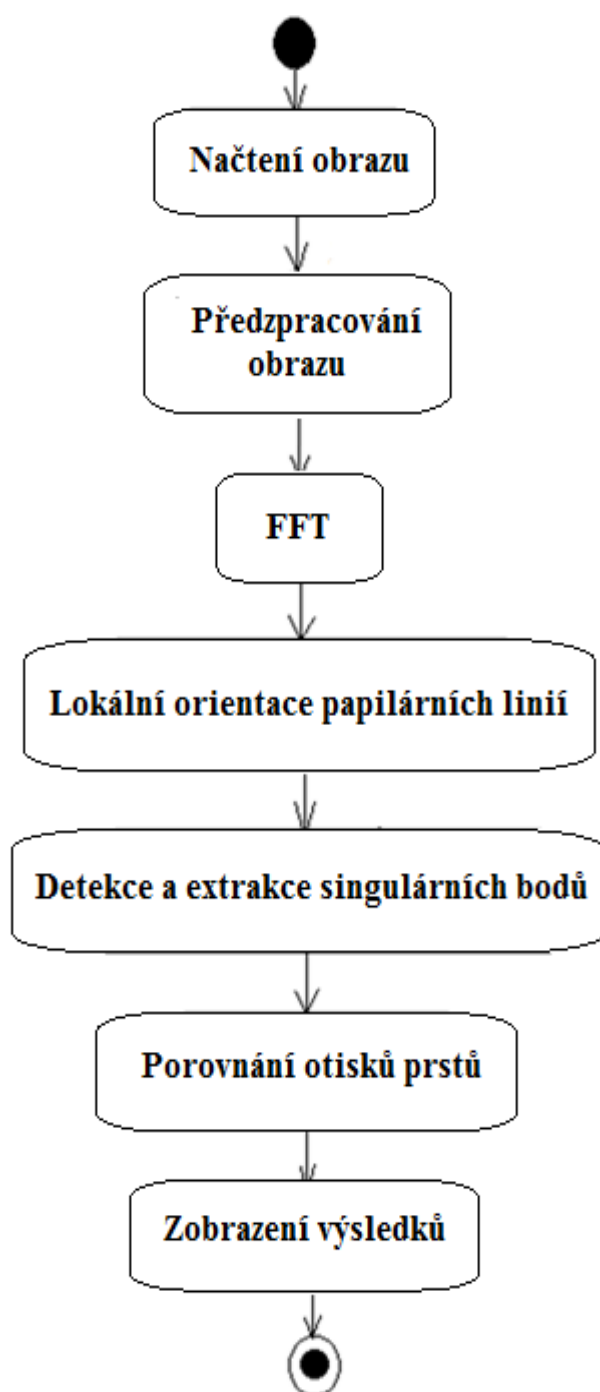
Msgbox – tento příkaz slouží k vytvoření dialogového okna se zprávou.

Imshow – tento příkaz slouží k zobrazení obrazu.

Numel – pomocí tohoto příkazu se zjistí, kolik prvků vstupní matice obsahuje.

Corr2 – tento příkaz slouží k výpočtu 2-D korelačního koeficientu.

3.3 Diagram aktivit softwaru pro identifikaci otisků prstů



Obr. 22: Diagram algoritmu pro porovnání otisků prstů

3.4 Načtení obrazu

Tento příkaz zobrazí pouze soubory s příponou *.jpg* .

```
[filename,path]=uigetfile({'*.jpg'});
```

V tomto případě příkaz načte grafický soubor, jenž je uložen v proměnné *filename*. Proměnná *path* obsahuje cestu, kde se nachází grafický soubor.

```
Image = imread(fullfile(path,filename));
```

3.5 Předzpracování obrazu

Jelikož vstupní obraz je ve formátu RGB, musí se nejprve pomocí příkazu *rgb2gray* převést do formátu GRAY (formát odstínu šedi).

```
Image = rgb2gray(Image);
```

Na obr. 23 je možné zpozorovat, že daný obraz nemá stejnoměrnou distribuci osvětlení, což může způsobit zhoršení kvality detekce singulárních bodů. Proto se pomocí příkazu *histeq* zvýší intenzita kontrastu, viz.: obr. 24.

```
Image= histeq(Image);
```



Obr. 23: Obraz před ekvalizací histogramem **Obr. 24:** Obraz po ekvalizaci histogramem

3.6 Zvýšení kvality obrazu pomocí Fourierovy transformace

Jak už bylo psáno výše, Fourierova transformace v tomto algoritmu, působí jako filtr dolní propusti, který filtruje vysoké frekvence. V tomto případě, vysokými frekvencemi, je na mysli šum, který zhoršuje kvalitu obrazu a převážně zvyšuje obtížnost detekce singulárních bodů. Dále také díky Fourierovy transformaci dochází k výraznému zlepšení kvality lokální orientace a směrového pole papilárních linií.

```
I=double(image);  
[w,h]=size(I);  
FFT_image=zeros(w,h);
```

Tento algoritmus pracuje tak, že se obraz rozdělí do jednotlivých bloků $W \times W$ (numerická hodnota velikosti W se pomocí experimentálních měření rozhodla na 80). Na tyto jednotlivé bloky se provedou příkazy, které jsou napsány podle rovnic (2.1) – (2.3). Nakonec se provede ekvalizace obrazu pro zvýšení kvality výstupního obrazu (viz.: Obr. 26).

```
for i=1:W:w  
    for j=1:W:h  
        if i+W-1 <= w && j+W-1 <= h  
            F=fft2(I(i:i+W-1,j:j+W-1));  
            factor=abs(F).^f;  
            reverse_f=abs(ifft2(F.*factor));  
            Max_reverse_f=max(reverse_f(:));  
  
            reverse_f=reverse_f./Max_reverse_f;  
            FFT_image(i:i+W-1,j:j+W-1)=reverse_f;  
        end  
    end  
end  
  
final=FFT_image*255;  
final=uint8(final);  
final=histeq(final);
```



Obr. 25: Vstupní obraz před provedením FFT **Obr. 26:** Výstupní obraz po provedení FFT

3.7 Lokální orientace papilárních linií

Jak už bylo popsáno výše, odhad lokální orientace slouží ke stanovení směru papilárních linií. Tato lokální orientace je nutná k detekci a následné extrakci singulárních bodů.

Popis algoritmu lokální orientace papilárních linií[18] :

a) Výpočet gradientů $G_x(x, y)$ a $G_y(x, y)$ za pomoci Sobelovy konvoluční masky.

Nejprve se pomocí příkazu *fspecial* vytvoří Sobelova konvoluční maska, viz.: obr. 15. Poté se pomocí dvoudimenzionálního **FIR** filtru vytvoří gradient osy x (horizontální směr). Aby se vytvořil gradient osy y (vertikální směr), je nutné pomocí příkazu *transpose* transponovat Sobelovu konvoluční masku. Nakonec se opět pomocí dvoudimenzionálního **FIR** filtru vytvoří gradient osy y.

```
gradient_filter=fspecial('sobel');
Gx = filter2-gradient_filter,image);
gradient_filter = transpose-gradient_filter);
Gy = filter2-gradient_filter,image);
```

b) Odhad lokální orientace obrazu Image (x, y)[23] :

Algoritmus byl vytvořen dle rovnic (2.4) – (2.7).

```
Gxy = Gx.*Gy;
Minus = (Gy - Gx).*(Gy + Gx);

blocksigma = 5;
velikost = fix(6*blocksigma);
if ~mod(velikost,2);    velikost = velikost+1; end
f = fspecial('Gaussian', velikost, blocksigma);
Minus = filter2(f, Minus);
Gxy = 2*filter2(f, Gxy);

noimgim = pi/2 + atan2(Gxy,Minus)/2;
```

- c) Následně kvůli šumu, přerušených hřebenů papilárních linií a dalším poškozením v obraze, bude provedena filtrace filtrem typu dolní propust. Proto, aby byla provedena filtrace typu dolní propust, musí být obraz převeden do spojitého vektorového pole:

```
FIx = cos(2*noimgim);
FIy = sin(2*noimgim);

maska = fspecial('gaussian',5);
```

Příkaz **imfilter** provede multidimenzionální filtraci vstupní proměnné **FIx** pomocí multidimenzionálního filtru **maska**, neboli bude provedena filtrace filtrem typu dolní propusti, kterým je **maska**. Tato dolní propust byla vytvořena pomocí příkazu **fspecial('Gaussian')**

```
gfx = imfilter(FIx, maska, 'symmetric', 'same');
gfy = imfilter(FIy, maska, 'symmetric', 'same');
```

d) Lokální orientace hřebenů papilárních linií v bodě (x, y)[23] :

Lokální orientace bude vypočtena pomocí rovnice (2.12).

```
noimg = atan2(gfy,gfx);
noimg(noimg<0) = noimg(noimg<0)+2*pi;
noimg = 0.5*noimg;
```

3.8 Detekce a extrakce singulárních bodů

V tomto případě se nejprve udělá segmentace obrazu, jenž obsahuje informace o směru daného obrazu. Tato informace je v radiánech, ale pro zjednodušení se převede na stupně. Po segmentaci se vytvoří binární obraz (a), který obsahuje informaci o tom, kde se nachází hodnoty stupně větší jak 150. Tento binární obraz (a) je důležitý v tom, že dává informaci, kde se nachází singulární bod **delta** a singulární bod “jednoduché” **jádro**. Následně se očísloví jednotlivé hodnoty v obraze, přičemž se následně za pomoci jednotlivých funkcí spočte, kde se nachází největší počet stejných čísel (hodnot - 1). Tam, kde se nachází největší počet čísel, se rovněž nachází singulární bod **delta** a singulární bod “jednoduché” **jádro**. Díky toho se vytvoří algoritmus, který nalezne tyto singulární body.

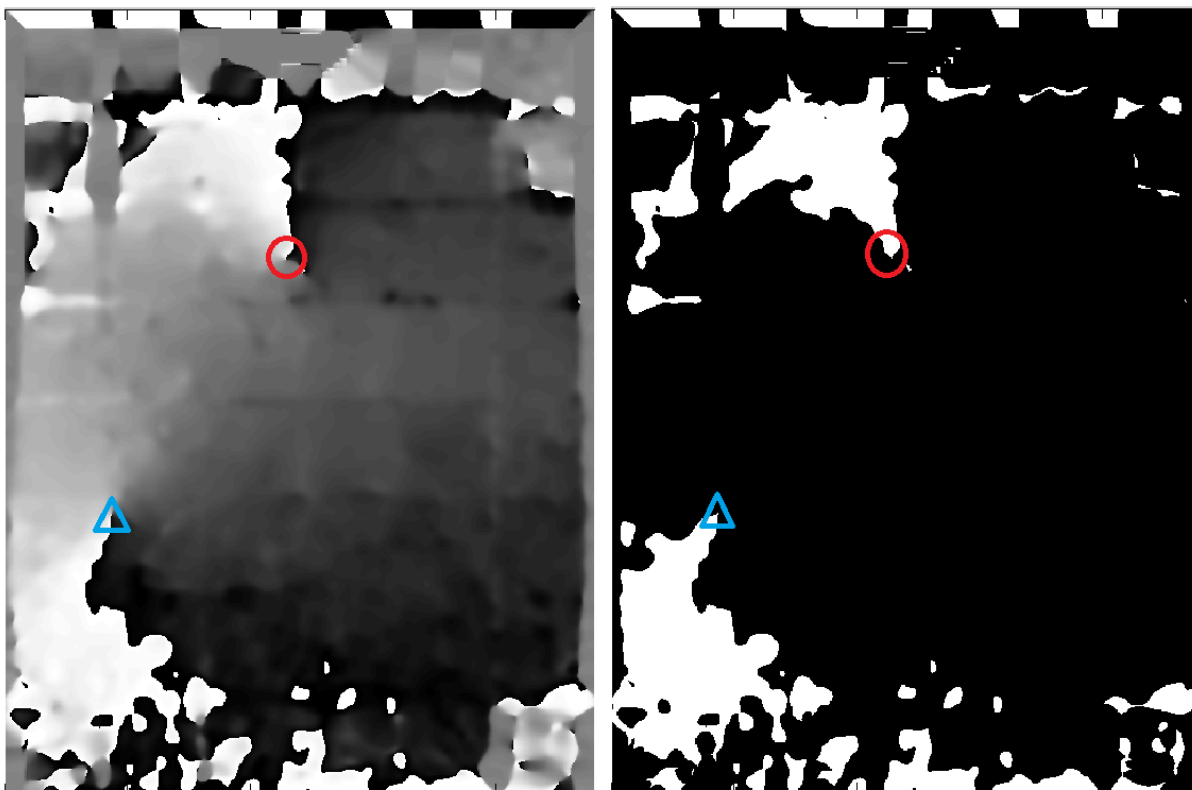
Popis algoritmu detekce a extrakce bodů jádro:

- a) **Převedení hodnot pixelů, jež jsou v radiánech na stupně (0 – 180°).**

```
noimgStupen = noimg.*180/pi;
```

- b) **Segmentace směrového pole**

```
a = noimgStupen > 150;
```



Obr. 27: Směrové pole před a po segmentaci

Na obr. 27 je možné vidět směrové pole před a po segmentaci, kde červeným kolečkem je znázorněn bod jádro a modrým trojúhelníkem zase bod delta. Z obrazu na levé straně je možné vyvodit, že singulární bod “**jednoduché**“ **jádro** a singulární bod delta se nacházejí tam, kde dochází k přelomu hodnot minima a maxima. Toto tvrzení popisují rovnice (2.13) – (2.15). Díky tomuto tvrzení je možné vyvodit, že tam kde dochází k přelomu mezi hodnotou 180 (maximální hodnota – vyjádřena bílou barvou) a 0 (minimální hodnota – vyjádřena černou barvou) se nachází singulární bod “**jednoduché**“ **jádro** nebo singulární bod delta. Proto se provede segmentace směrového pole, tato segmentace je znázorněna na pravé straně obr. 27.

Na segmentovaném obraze v obr. 27 je možné vidět, že obraz obsahuje strašně hodně takových přelomů, kde se může a nemusí nacházet singulární bod “**jednoduché**“ **jádro** a singulární bod delta. Proto musí dojít k další segmentaci. U této segmentace se ponechají ty úseky, jenž mají největší zastoupení jedniček. Aby se zjistilo, které bloky mají největší zastoupení jedniček, použije se příkaz ***bwlabel***, viz.: níže.

```
Ilabeled2 = bwlabel(a);
```



Obr. 28: Směrové pole po zavedení indexu

Tento příkaz ***bwlabel*** slouží k označení jednotlivých úseků za pomoci číselných indexů. Po zobrazení výstupní proměnné ***Ilabel2*** dojde k barevnému odlišení jednotlivých úseků, což napomůže při závěrečné kontrole.

Pomocí příkazu ***max*** se zjistí maximální hodnoty indexů, jež obsahuje obr. 28.

```
max_label = max(max(Ilabel2));
```

Příkazy v cyklu ***for*** (viz.: níže) slouží k vyhledání největšího zastoupení úseku a to tak, že nejprve se pomocí příkazu ***find*** naleznou pozice jednotlivých indexů v daných úsecích. Dále se spočte příkazem ***length*** kolik daný úsek obsahuje takovýchto pozic.

```
label = zeros(max_label,1);
for k = 1:max_label
    [x ~] = find(Ilabel2 == k);
    label(k,1) = length(x);
end
```

Poté pomocí příkazu ***sort*** se seřadí čísla od největšího počtu po nejmenší. První dvě čísla v proměnné ***index*** obsahují úseky s největšími zastoupeními jedniček. Právě v těchto dvou úsecích se nalézá bod jádro a delta.

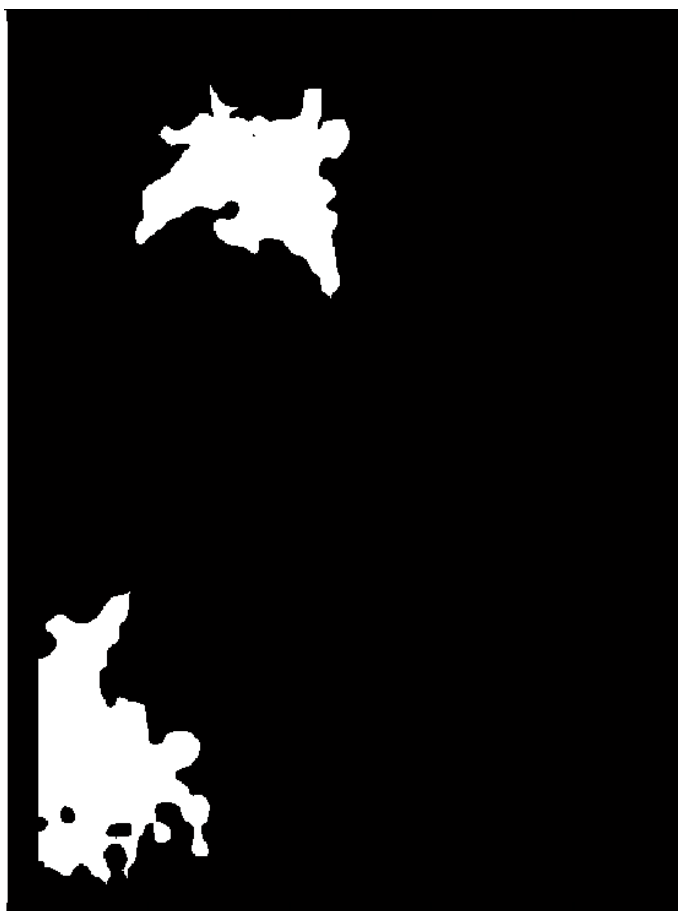
```
[~, labely] = sort(label, 'descend');
index = labely(1:2,1);
```

Dále v tomto cyklu se hledají úseky, jimž patří první dvě čísla v indexu.

```
for i= (n+1+15):(w+n-15)
    for j=(n+1+15):(h+n-15)
        if Ilabel2(i,j) == index(1,1) || Ilabel2(i,j) == index(2,1) %||
            Ilabel2(i,j) == index(3,1)
                labla(i,j) = 1;
            end
        end
    end
```

Příklad:

Na obr. 28 pod číslem **1**, se nalézá úsek jedniček, které mají index 30. Takovýchto jedniček s indexem 30 v tomto úseku je např. **5000**. V dalším úseku pod číslem **5** jich bude **500**, v úseku pod číslem **3** jich bude **2000** atd. Tyto čísla se pak seřadí od největšího po nejmenší. A první dvě čísla obsahují úseky s největšími zastoupeními jedniček. Dále se pomocí jednotlivých příkazů přiřadí dané čísla k jednotlivým úsekům. Např. se zjistí, že první dvě největší čísla jsou 5000 a 2000... tyto dvě čísla patří úsekům pod číslem 1 a 3. To znamená, že bod jádro a delta jsou v úseku 1 a 3 (viz.: obr. 29).



Obr. 29: Směrové pole po provedení druhé segmentace

Na obr. 29 je možné vidět dva úseky, ve kterých se nachází singulární body jádro a delta.

- c) Pomocí masky okolí se bude hledat singulární bod “jednoduché” jádro, za pomoci algoritmu:

Algoritmus níže je napsán podle rovnice (2.13) s tím rozdílem, že velikost masky okolí není (3 x 3), ale (9 x 9) a to z důvodu lepší kvality vyhodnocení.

Dále pokud bod $\text{noimgStupen}(i-4, j+3)$ (noimgStupen je směrové pole před segmentací neboli také obr. 17) se bude rovnat hodnotě $\text{labla}(i-4, j+3) = 0$ (obraz labla je na obr. 29) – nejedná se o singulární bod “jednoduché” jádro. Je to z důvodu toho, že algoritmus může nalézt několik singulárních bodů “jednoduché” jádro, což by způsobilo chybu. Proto se napsala podmínka, která zaručuje, že se vybere bod jádra, který se nalézá v jednom ze dvou úseků, o nichž se ví, že na 100 % obsahují singulární bod “jednoduché” jádro a singulární bod delta.


```

for i= (n+1+15):(w+n-15)
    for j=(n+1+15):(h+n-15)
if (noimgStupen(i-4,j+4)<noimgStupen(i-3,j+4)&& ...
noimgStupen(i-3,j+4)<noimgStupen(i-2,j+4)&& ...
noimgStupen(i-2,j+4)<noimgStupen(i-1,j+4)&& ...
noimgStupen(i-1,j+4)<noimgStupen(i ,j+4)&& ...
noimgStupen(i ,j+4)<noimgStupen(i+1,j+4)&&...
noimgStupen(i+1,j+4)<noimgStupen(i+2,j+4)&& ...
noimgStupen(i+2,j+4)<noimgStupen(i+3,j+4)&& ...
noimgStupen(i+3,j+4)<noimgStupen(i+4,j+4)&& ...
noimgStupen(i+4,j+4)<noimgStupen(i+4,j+3)&& ...
noimgStupen(i+4,j+3)<noimgStupen(i+4,j+2)&& ...
noimgStupen(i+4,j+2)<noimgStupen(i+4,j+1)&& ...
noimgStupen(i+4,j+1)<noimgStupen(i+4,j )&& ...
noimgStupen(i+4,j )<noimgStupen(i+4,j-1)&& ...
noimgStupen(i+4,j-1)<noimgStupen(i+4,j-2)&& ...
noimgStupen(i+4,j-2)<noimgStupen(i+4,j-3)&& ...
noimgStupen(i+4,j-3)<noimgStupen(i+4,j-4)&& ...
noimgStupen(i+4,j-4)<noimgStupen(i+3,j-4)&& ...
noimgStupen(i+3,j-4)<noimgStupen(i+2,j-4)&& ...
noimgStupen(i+2,j-4)<noimgStupen(i+1,j-4)&& ...
noimgStupen(i+1,j-4)<noimgStupen(i ,j-4)&& ...
noimgStupen(i ,j-4)<noimgStupen(i-1,j-4)&& ...
noimgStupen(i-1,j-4)<noimgStupen(i-2,j-4)&& ...
noimgStupen(i-2,j-4)<noimgStupen(i-3,j-4)&& ...
noimgStupen(i-3,j-4)<noimgStupen(i-4,j-4)&& ...
noimgStupen(i-4,j-4)<noimgStupen(i-4,j-3)&& ...
noimgStupen(i-4,j-3)<noimgStupen(i-4,j-2)&& ...
noimgStupen(i-4,j-2)<noimgStupen(i-4,j-1)&& ...
noimgStupen(i-4,j-1)<noimgStupen(i-4,j )&& ...
noimgStupen(i-4,j )<noimgStupen(i-4,j+1)&& ...
noimgStupen(i-4,j+1)<noimgStupen(i-4,j+2)&& ...
noimgStupen(i-4,j+2)<noimgStupen(i-4,j+3)&& labla(i-4,j+3) ~= 0)
        core_label(i,j) = 1;
    end
    end;
end;

```

d) V případě, že se jedná o vzor závit (whorl), hledá se jádro i pro dolní smyčku

Algoritmus níže je napsán podle rovnice (2.14) s tím rozdílem, že velikost masky okolí není (3 x 3), ale (9 x 9) a to z důvodu lepší kvality vyhodnocení.

```

for i= (n+1+15):(w+n-15)
    for j=(n+1+15):(h+n-15)
        if (noimgStupen(i+4,j-3)>noimgStupen(i+4,j-2) && ...
noimgStupen(i+4,j-2)>noimgStupen(i+4,j-1) && ...
noimgStupen(i+4,j-1)>noimgStupen(i+4,j) && ...
noimgStupen(i+4,j) >noimgStupen(i+4,j+1) && ...
noimgStupen(i+4,j+1)>noimgStupen(i+4,j+2) && ...
noimgStupen(i+4,j+2)>noimgStupen(i+4,j+3) && ...
noimgStupen(i+4,j+4)>noimgStupen(i+3,j+4) && ...
noimgStupen(i+3,j+4)>noimgStupen(i+2,j+4) && ...
noimgStupen(i+2,j+4)>noimgStupen(i+1,j+4) && ...
noimgStupen(i+1,j+4)>noimgStupen(i,j+4) && ...
noimgStupen(i,j+4)>noimgStupen(i-1,j+4) && ...
noimgStupen(i-1,j+4)>noimgStupen(i-2,j+4) && ...
noimgStupen(i-2,j+4)>noimgStupen(i-3,j+4) && ...
noimgStupen(i-3,j+4)>noimgStupen(i-4,j+4) && ...
noimgStupen(i-4,j+4)>noimgStupen(i-4,j+3) && ...
noimgStupen(i-4,j+3)>noimgStupen(i-4,j+2) && ...
noimgStupen(i-4,j+2)>noimgStupen(i-4,j+1) && ...
noimgStupen(i-4,j+1)>noimgStupen(i-4,j) && ...
noimgStupen(i-4,j) >noimgStupen(i-4,j-1) && ...
noimgStupen(i-4,j-1)>noimgStupen(i-4,j-2) && ...
noimgStupen(i-4,j-2)>noimgStupen(i-4,j-3) && ...
noimgStupen(i-4,j-3)>noimgStupen(i-4,j-4) && ...
noimgStupen(i-4,j-4)>noimgStupen(i-3,j-4) && ...
noimgStupen(i-3,j-4)>noimgStupen(i-2,j-4) && ...
noimgStupen(i-2,j-4)>noimgStupen(i-1,j-4) && ...
noimgStupen(i-1,j-4)>noimgStupen(i,j-4) && ...
noimgStupen(i,j-4)>noimgStupen(i+1,j-4) && ...
noimgStupen(i+1,j-4)>noimgStupen(i+2,j-4) && ...
noimgStupen(i+2,j-4)>noimgStupen(i+3,j-4) && ...
noimgStupen(i+3,j-4)>noimgStupen(i+4,j-4))
            coreDouble_label(i,j) = 1;

```

e) Nalezení bodu delta za pomoci rovnice:

Algoritmus níže je napsán podle rovnice (2.15) s tím rozdílem, že velikost masky okolí není jako v předchozích případech (9 x 9), ale (15 x 15).

```

for i= (n+1+15):(w+n-15)
    for j=(n+1+15):(h+n-15)
        if (noimgStupen(i+7,j-6)<noimgStupen(i+7,j-5) && ...
noimgStupen(i+7,j-5)<noimgStupen(i+7,j-4) && ...

```

```

noimgStupen(i+7,j-4)<noimgStupen(i+7,j-3) && ...
noimgStupen(i+7,j-3)<noimgStupen(i+7,j-2) && ...
noimgStupen(i+7,j-2)<noimgStupen(i+7,j-1) && ...
noimgStupen(i+7,j-1)<noimgStupen(i+7,j) && ...
noimgStupen(i+7,j)<noimgStupen(i+7,j+1) && ...
noimgStupen(i+7,j+1)<noimgStupen(i+7,j+2) && ...
noimgStupen(i+7,j+2)<noimgStupen(i+7,j+3) && ...
noimgStupen(i+7,j+3)<noimgStupen(i+7,j+4) && ...
noimgStupen(i+7,j+4)<noimgStupen(i+7,j+5) && ...
noimgStupen(i+7,j+5)<noimgStupen(i+7,j+6) && ...
noimgStupen(i+7,j+7)<noimgStupen(i+6,j+7) && ...
noimgStupen(i+6,j+7)<noimgStupen(i+5,j+7) && ...
noimgStupen(i+5,j+7)<noimgStupen(i+4,j+7) && ...
noimgStupen(i+4,j+7)<noimgStupen(i+3,j+7) && ...
noimgStupen(i+3,j+7)<noimgStupen(i+2,j+7) && ...
noimgStupen(i+2,j+7)<noimgStupen(i+1,j+7) && ...
noimgStupen(i+1,j+7)<noimgStupen(i,j+7) && ...
noimgStupen(i,j+7)<noimgStupen(i-1,j+7) && ...
noimgStupen(i-1,j+7)<noimgStupen(i-2,j+7) && ...
noimgStupen(i-2,j+7)<noimgStupen(i-3,j+7) && ...
noimgStupen(i-3,j+7)<noimgStupen(i-4,j+7) && ...
noimgStupen(i-4,j+7)<noimgStupen(i-5,j+7) && ...
noimgStupen(i-5,j+7)<noimgStupen(i-6,j+7) && ...
noimgStupen(i-6,j+7)<noimgStupen(i-7,j+7) && ...
noimgStupen(i-7,j+7)<noimgStupen(i-7,j+6) && ...
noimgStupen(i-7,j+6)<noimgStupen(i-7,j+5) && ...
noimgStupen(i-7,j+5)<noimgStupen(i-7,j+4) && ...
noimgStupen(i-7,j+4)<noimgStupen(i-7,j+3) && ...
noimgStupen(i-7,j+3)<noimgStupen(i-7,j+2) && ...
noimgStupen(i-7,j+2)<noimgStupen(i-7,j+1) && ...
noimgStupen(i-7,j+1)<noimgStupen(i-7,j) && ...
noimgStupen(i-7,j)<noimgStupen(i-7,j-1) && ...
noimgStupen(i-7,j-1)<noimgStupen(i-7,j-2) && ...
noimgStupen(i-7,j-2)<noimgStupen(i-7,j-3) && ...
noimgStupen(i-7,j-3)<noimgStupen(i-7,j-4) && ...
noimgStupen(i-7,j-4)<noimgStupen(i-7,j-5) ...
noimgStupen(i-7,j-5)<noimgStupen(i-7,j-6) && ...
noimgStupen(i-7,j-6)<noimgStupen(i-7,j-7) && ...
noimgStupen(i-7,j-7)<noimgStupen(i-6,j-7) && ...
noimgStupen(i-6,j-7)<noimgStupen(i-5,j-7) && ...
noimgStupen(i-5,j-7)<noimgStupen(i-4,j-7) && ...
noimgStupen(i-4,j-7)<noimgStupen(i-3,j-7) && ...
noimgStupen(i-3,j-7)<noimgStupen(i-2,j-7) && ...

```

```

noimgStupen(i-2,j-7)<noimgStupen(i-1,j-7) && ...
noimgStupen(i-1,j-7)<noimgStupen(i ,j-7) && ...
noimgStupen(i ,j-7)<noimgStupen(i+1,j-7) && ...
noimgStupen(i+1,j-7)<noimgStupen(i+2,j-7) && ...
noimgStupen(i+2,j-7)<noimgStupen(i+3,j-7) && ...
noimgStupen(i+3,j-7)<noimgStupen(i+4,j-7) && ...
noimgStupen(i+4,j-7)<noimgStupen(i+5,j-7) && ...
noimgStupen(i+5,j-7)<noimgStupen(i+6,j-7) && ...
noimgStupen(i+6,j-7)<noimgStupen(i+7,j-7) )
        delta_label(i,j) = 1;

```

3.9 Porovnání otisků prstů pomocí singulárních bodů

Nejprve se naleznou jedničky pro singulární body **delta**, “**jednoduché**“ **jádro** a “**dvojité**“ **jádro** (v případě vzoru závit nebo dvojitá smyčka). Následně se vypočte medián *x* a *y* hodnoty jednoduché jádro. Tyto hodnoty jsou následně použity v algoritmu, jenž rozhoduje, zdali existuje singulární bod “**dvojité**“ **jádro** nebo **dvojitá smyčka**.

Tento algoritmus pracuje tak, že jestliže proměnná **DCx** není nulová, tak se spočítá vzdálenost singulárního bodu “**dvojité**“ **jádro**. To znamená, že se spočte vzdálenost mezi singulárním bodem **jádro** horní a spodní smyčky. Pokud vzdálenost bude menší, jak 150 pixelů a zároveň singulární bod **jádro** dolní smyčky se bude nacházet níže, než singulární bod **jádro** horní smyčky, tak jsou tyto body považovány za reálný singulární bod “**dvojité**“ **jádro**. V opačném případě neexistuje singulární bod „**dvojité**“ **jádro**.

Dále program obsahuje rozhodovací algoritmy, které počítají vzdálenost mezi singulárním bodem “**jednoduché**“ **jádro** a singulárním bodem **delta** nebo singulárním bodem “**dvojité**“ **jádro** a bodem **delta**. Výstupem tohoto celého algoritmu je proměnná **distance**, jež představuje onu vzdálenost mezi body a 3 řádkovou proměnnou, ve které je zakódována informace, zdali obraz obsahuje singulární bod **delta** a singulární bod “**dvojité**“ **jádro**.

Popis algoritmu porovnání otisků prstů:

- a) **Detekce jedniček pro singulární body delta, jednoduché jádro a dvojité jádro (v případě vzoru závit nebo dvojitá smyčka)**

```

[core_labelx core_labely]=find(core_label == 1); % nalezení CORE1
core_labelx = round(median(core_labelx)); % vysledek COREx
core_labely = round(median(core_labely)); % vysledek COREy
[delta_labelx delta_labely]=find(delta_label == 1); % nalezení DELTA1
[DCx DCy]=find(coreDouble_label == 1); % nalezení Double CORE

```

b) Dále se zjistí, zdali dvojité jádro, které bylo detekováno, opravdu existuje.

Algoritmus pracuje, tak že se nejprve pomocí podmínky **if** zjistí, zdali proměnná **DCx** je prázdná. Jestliže je prázdná, obraz neobsahuje singulární bod “**dvojité**” **jádro**, což znamená, že se nejedná o otisk typu **závit** (whorl) nebo **dvojitá smyčka** (twin loop).

Pokud proměnná **DCx** není prázdná, vypočte se euklidovská vzdálenost mezi singulárním bodem **jádro** horní a spodní smyčky pomocí rovnice (2.16). Až se vypočte vzdálenost mezi těmito dvěma singulárními body, vypíše se podmínka, po které se rozhodne, zdali se jedná o opravdový singulární bod “**dvojité**” **jádro**. To znamená, že pokud vzdálenost singulárního bodu **jádro** horní a spodní smyčky je menší, než 150 pixelů a zároveň poloha singulárního bodu **jádro** spodní smyčky se nalézá níže, než singulární bod **jádro** horní smyčky – existuje singulární bod “**dvojité**” **jádro** a zároveň se jedná o otisk typu **závit** (whorl) nebo **dvojitá smyčka** (twin loop).

```
if ~isempty(DCx)

    for k = 1:length(DCx)
        DistD2Core(k) = round(sqrt((core_labelx(1,1)-DCx(k))^2 ...
                                   +(core_labely(1,1)-DCy(k))^2));
    end
    for k = 1:length(DCx)
        if DistD2Core(k) < 150 && DCx(k) > core_labely(1,1)
            Double_core = [DCx(k) DCy(k)];
        end
    end
end
```

c) Jestliže obraz obsahuje jeden singulární bod “jednoduché” jádro a delta

Jestliže obraz obsahuje jeden singulární bod “**jednoduché**” **jádro** a **delta**, jedná se otisk typu **pravá** nebo **levá smyčka**. Nejprve se vypočte medián singulárního bodu **delta** pro souřadnice **x** a **y**.

```
if size(Double_core,1)==0 && size(delta_labelx,1)>0
    delta_labelx = round(median(delta_labelx));
    delta_labely = round(median(delta_labely));
```

Dále se vypočte euklidovská vzdálenost mezi singulárním bodem “**jednoduché**” **jádro** a bodem **delta**.

```
Distance = round(sqrt((core_labelx-delta_labelx)^2 ...
                      +(core_labely-delta_labely)^2));
```

Tento cyklus stejně jako v předchozím případě slouží k rozeznání mezi pravou a levou smyčkou. V případě, že výsledek podmínky **if** bude záporný, jedná se o otisk typu levá smyčka, naopak pokud výsledek bude kladný, jedná se o otisk typu pravá smyčka.

```
if (Double_core(1,2) - delta_labely) < 0
    Strana = 1; % leva
else
    Strana = 5; % prava
end
```

Následně se vypočte vzdálenost osy y, což je rozdíl y-ových souřadnic bodu “**jednoduché**” **jádro** a **delty**. S euklidovské vzdálenosti a vzdálenosti y-ové osy se vypočte úhel. Hodnota úhlu se poté převede s radiánu na stupně a následně použije při závěrečném porovnání otisků prstů.

```
y = abs(core_labely - delta_labely)
Uhel = y/Distance;
Uhel = sin(Uhel);
Uhel = Uhel.*180/pi;
```

Nakonec se do prvního řádku proměnné **Check** uloží hodnota 1, což znamená, že se jedná o **pravou** nebo **levou smyčku** s jedním singulárním bodem “**jednoduché**” **jádro** a bodem **delta**.

```
Check(1) = 1;
```

d) Jestliže obraz obsahuje singulární bod “**dvojité**” **jádro** a bod **delta**.

Opětne se vypočte medián singulárního bodu **delta** pro souřadnice x a y.

```
elseif size(Double_core,1)>0 && size(delta_labelx,1)>0

delta_labelx = round(median(delta_labelx));
delta_labely = round(median(delta_labely));
```

Dále se, jako v předchozím případě, vypočte euklidovská vzdálenost s tím rozdílem, že se vzdálenost bude počítat mezi singulárním bodem “**dvojité**” **jádro** a bodem **delta**.

```
Distance = round(sqrt((Double_core(1,1)-delta_labelx)^2 ...
                    +(Double_core(1,2)-delta_labely)^2));
```

Následně se vypočte vzdálenost osy y, a to mezi souřadnicemi singulárního bodu **dvojité jádro** a **delty**. S euklidovské vzdálenosti a vzdálenosti y-ové osy se vypočte úhel. Tato hodnota úhlu se opět převede s radiánu na stupně a použije při závěrečném porovnání otisků prstů.

```
y = abs(Double_core(1,2)-delta_labely);
Uhel = y/Distance;
Uhel = sin(Uhel);
```

```
Uhel = Uhel.*180/pi;
```

Nakonec se do třetího řádku proměnné **Check** uloží hodnota 1, což znamená, že se jedná o obraz se dvěma smyčkami, např. závit, přičemž první smyčka je singulární bod jednoduché jádro (horní půlkruh) a druhá smyčka je singulární bod dvojité jádro (dolní půlkruh).

```
Check(3) = 1;
```

e) Algoritmus porovnání otisků prstů

Nejprve se vypočte rozdíl vzdáleností a uhlů jednotlivých otisků prstů.

```
Delta_Distance = abs(Distance2 - Distance1)
```

```
Delta_uhel = abs(Uhel2 - Uhel1)
```

Dále se vypočte korelační koeficient směrového pole jednotlivých otisků prstů.

```
korelace_noimg = corr2(noimgStupen1,noimgStupen2);
```

Následně se podmínky **if** vypíše, zdali se proměnná **Check1** (proměnná pro první otisk prstu) a **Check2** (proměnná pro druhý otisk prstu) shoduje. Zároveň se rozhoduje, zdali rozdíl vzdálenosti je menší, než 5 pixelů a rozdíl uhlů je menší, než 8 pixelů. Jestliže bude tato podmínka splněna, jedná se o dva stejné otisky prstu, pokud nebude, přejde se na další podmínku.

Pokud se proměnná **Check1** (proměnná pro první otisk prstu) a **Check2** (proměnná pro druhý otisk prstu) shoduje a rozdíl vzdálenosti je menší, než 5 pixelů a menší, než 20 pixelů ($5 < \text{rozdíl vzdálenosti} < 20$), a zároveň korelační koeficient bude mít víc, jak 40,55 %, rozdíl úhlu nebude větší, než 8° - jedná se o dva stejné otisky prstu

Výstupem tohoto algoritmu je dialogové okno, které pomocí příkazu **msgbox** informuje, zdali daný otisk prstu je stejný nebo ne.

```
if ((Check1 == Check2) & (Delta_Distance <= 5) & (Delta_uhel < 8) &
(Strana1 == Strana2))
    msgbox('OTISK JE STEJNÝ','Verify system','custom',icon)
elseif ((Check1 == Check2) & (Delta_Distance <= 20) & (Delta_uhel <
8) ...
    & (Strana1 == Strana2) & (korelace_noimg > 0.4055))
    msgbox('OTISK JE STEJNÝ','Verify system','custom',icon)
else
    msgbox('Chyba: Otisk není stejný','ERROR','warn');
end
```

3.10 Zobrazení výsledků

Pro zobrazení výsledků slouží příkaz *regionprops*, který měří vlastnosti spojených objektů. Jako spojený objekt slouží proměnná **Ilabel**, která obsahuje matici nulových hodnot o velikosti obrazu otisku prstu. Do této matice se uloží hodnota 1. Pozice, kde se tato hodnota 1 uloží, udává souřadnice singulárního bodu. To znamená, že pokud singulární bod **delta** bude mít souřadnice [300, 300], tak na této pozici v proměnné **Ilabel** bude 1, všude jinde bude hodnota 0. Jelikož příkaz *regionprops*, jak už bylo napsáno, měří vlastnosti spojených objektů, bude měřit vlastnosti jediného objektu, což je hodnota 1 na pozici [300, 300]. Výstup příkazu *regionprops* je proměnná **stat_core** (**stat_double**).

Tato proměnná se dále použije při zobrazení singulární bodů, k tomu slouží příkaz *plot*, který vyobrazí jednotlivé kružnice singulárních bodů. Příklad toho, jak to vypadá – viz.: obr. 30.

```
Ilabel_core(core_labelx,core_labely) = 1;

stat_core = regionprops(Ilabel_core,'centroid');
stat_double = regionprops(Ilabel_double,'centroid');
```

Příkaz *imshow* slouží pro zobrazení obrazu.

```
imshow(o); hold on;

for x = 1: numel(stat_core)
    plot(stat_core(x).Centroid(1),stat_core(x).Centroid(2),'ro','LineWidth',5);
    hold on;
end

for x = 1: numel(stat_double)
    plot(stat_double(x).Centroid(1),stat_double(x).Centroid(2),'ro','LineWidth',5);
end
```

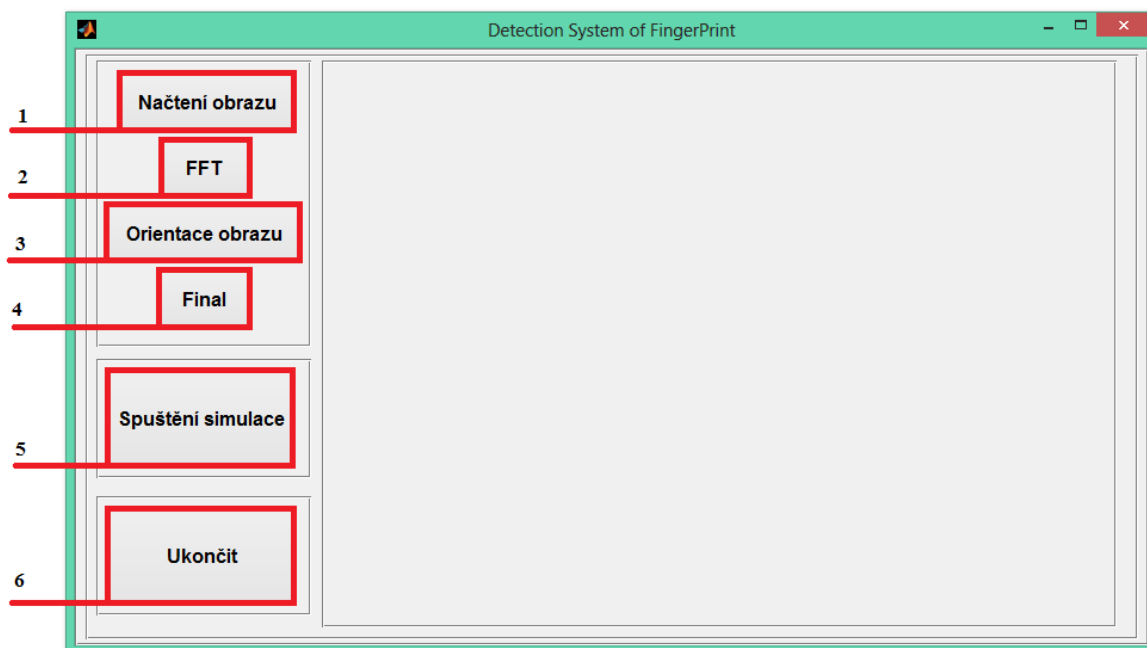



Obr. 30: Obráz otisku prstu s vyznačenými singulárními body

Na obr. 30 je možné vidět dva singulární body - **“jednoduché“ jádro** a **delta**. Singulární bod **“jednoduché“ jádro** je označen červeně, singulární bod **delta** modře.

3.11 Vytvoření a popis uživatelského rozhraní GUI

Uživatelské rozhraní bylo vytvořeno bez pomoci GUIDE (Graphical User Interface Development Environment), což je grafické uživatelské rozhraní, které umožňuje vytvářet aplikace s grafickým rozhraním. Uživatelské rozhraní bylo vytvořeno pouze za pomoci jednoho souboru s příponou *.m (M-file), který obsahuje celý popis grafické části tak i kód, který ovládá GUI[19] ,[20] .



Obr. 31: Popis uživatelského rozhraní GUI

- 1) **Načtení obrazu** – pomocí tohoto tlačítka dojde k načtení jednotlivých otisků prstů. Nejprve se musí vybrat první porovnávaný otisk prstu a následně se vybere druhý.
- 2) **FFT** – toto tlačítko provede zvýšení kvality obrazu otisku prstu pomocí Fourierovy řady.
- 3) **Orientace obrazu** – toto tlačítko zobrazí směrové pole vstupního obrazu.
- 4) **Final** – toto tlačítko slouží pro následné porovnání jednotlivých otisků prstů
- 5) **Spuštění simulace** – toto tlačítko provede kroky 2 – 4 zároveň.
- 6) **Ukončit** – toto tlačítko slouží pro ukončení programu a následnému vymazání všech uložených proměnných.

4. Vizualizace a srovnání naměřených výsledků s teoretickými předpoklady

V této kapitole se budou srovnávat jednotlivé nasnímané otisky prstu. Vždy se vyberou dva otisky prstu, jež budou mezi sebou porovnávány. To znamená, že tento program slouží k současnému porovnávání dvou nasnímaných otisků prstů. Nejprve se nafotí vzorový otisk prstu a následně se nafotí otisk prstu, jenž bude s tímto vzorovým otiskem prstu porovnán.

Na výběr bude mezi čtyřmi typy otisky prstů – **pravá smyčka** (right loop), **levá smyčka** (left loop), **závit (whorl)** a **klenutý oblouk** (tented arch).

Nejprve se budou srovnávat jednotlivé otisky prstů, jež jsou shodné, a nakonec se provede zkouška systému se dvěma různými otisky prstů, přičemž by systém měl zareagovat chybovou hláškou.

Metoda, jež byla použita při snímání, je metoda kontaktní, přičemž se použil optický snímač. Jako osvětlovací prvek byla použita lampička, která měla matné sklo pro lepší distribuci světla na snímáný povrch. Bohužel se nesehnal potřebný optický hranol, místo toho bylo použito sklo se zakřivením, které při dostatečném natočení a správné poloze osvětlení, pracovalo na podobném principu jako optický hranol. Nakonec jako snímací prvek byla použita webová kamera **FaceCam 1010**.

Specifikace kamery:

- 720p HD, 1.3 M rozlišení
- Manuální ostření
- Uložení ve formátu .JPG/WMV
- Rozlišení 1280 x 1024, 1280 x 720, 640 x 480 pixelů (V této práci bylo použito rozlišení 640 x 480 pixelů, vyšší nebylo vybráno a to z důvodu dostačující kvality a také by vyšší rozlišení zbytečně zpomalilo výpočetní čas).
- Podpora – Windows 7/ Vista/ XP SP2, MAC 10.4.9 a vyš, Linux 2.6.21 a vyš.
- Propojení s PC přes USB.

Specifikace PC:

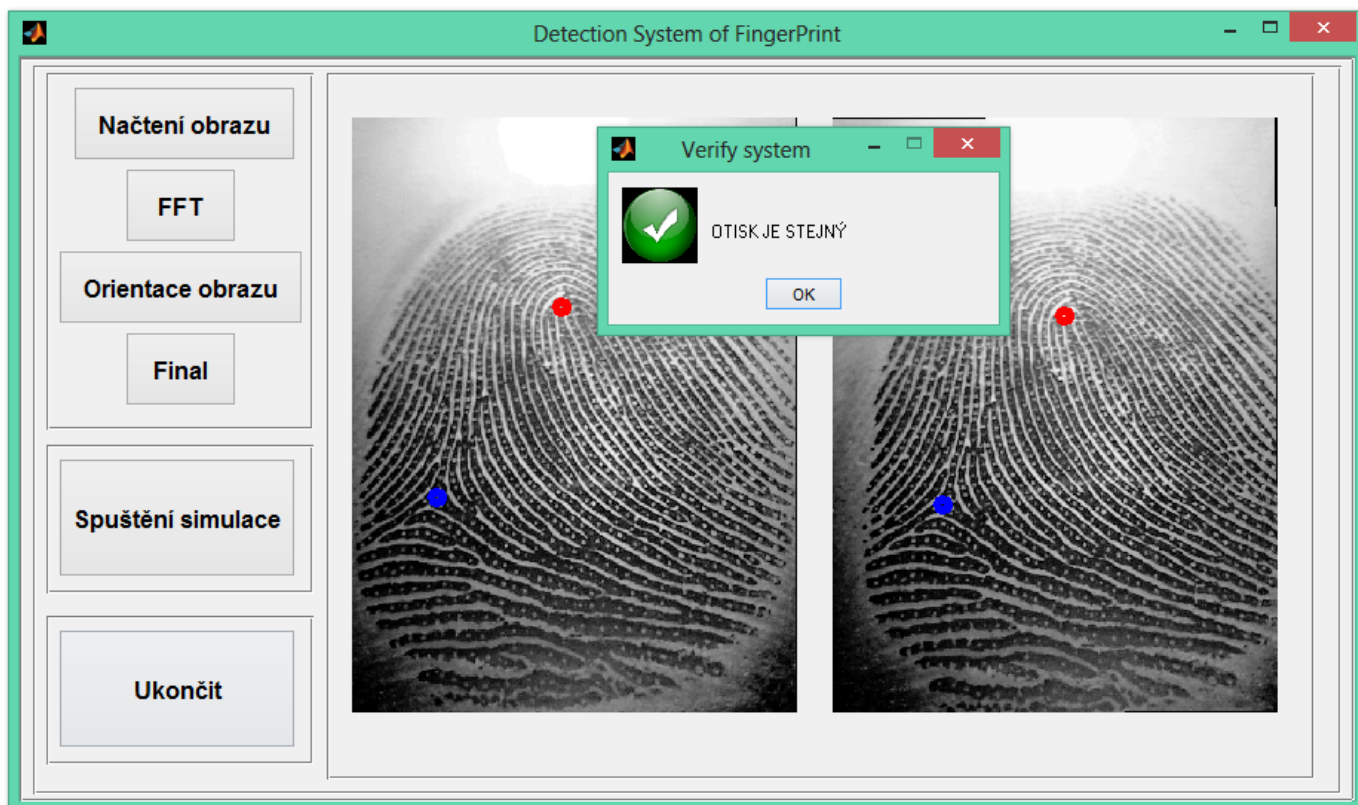
- Procesor: AMD Athlon(tm) 64 X2 Dual Core Processor 4000+ (2 CPU), ~2,1 GHz
- Paměť: 3072 MB DDR2
- Grafika: NVIDIA GeForce 8500 GT, 512 MB

4.1 Srovnání otisků prstů – pravá smyčka (right loop)

Na obr. 32, viz.: níže, je možné zpozorovat, že dané porovnávané otisky prstů jsou typu **pravá smyčka** (right loop) a to z důvodu toho, že obsahují singulární body typu **delta** a “**jednoduché**” **jádro**, přičemž singulární bod **delta** se nachází na levé straně a dále dochází k natočení smyčky doprava.

Tento program jak už bylo napsáno, slouží k současnému porovnávání dvou nasnímaných otisků prstů. Takže nejprve nafotí vzorový otisk prstu a následně se nafotí otisk prstu, jenž bude s tímto vzorovým porovnáván. Vzorový otisk je znázorněn na levé straně a porovnávaný na pravé.

Porovnávací algoritmus vyhodnotil, že rozdíl vzdáleností singulárního bodu “**jednoduché**” **jádru** a singulárního bodu **delta** je 2 pixely, což při maximálním možném rozdílu 20 pixelů, je totálně zanedbatelné. Dále také jelikož rozdíl vzdálenosti mezi singulárními body není větší, než 5 pixelů, není třeba dělat korelační koeficient směrového pole. Rozdíl úhlů jednotlivých otisků prstů činí 0.3849, maximální možný rozdíl je 8 stupňů, stejně jako u rozdílu vzdálenosti, tento předpoklad byl splněn. Jestliže byly tyto předpoklady splněny, program vyhodnotí porovnávané otisky prstů jako shodné, viz.: obr. 32.

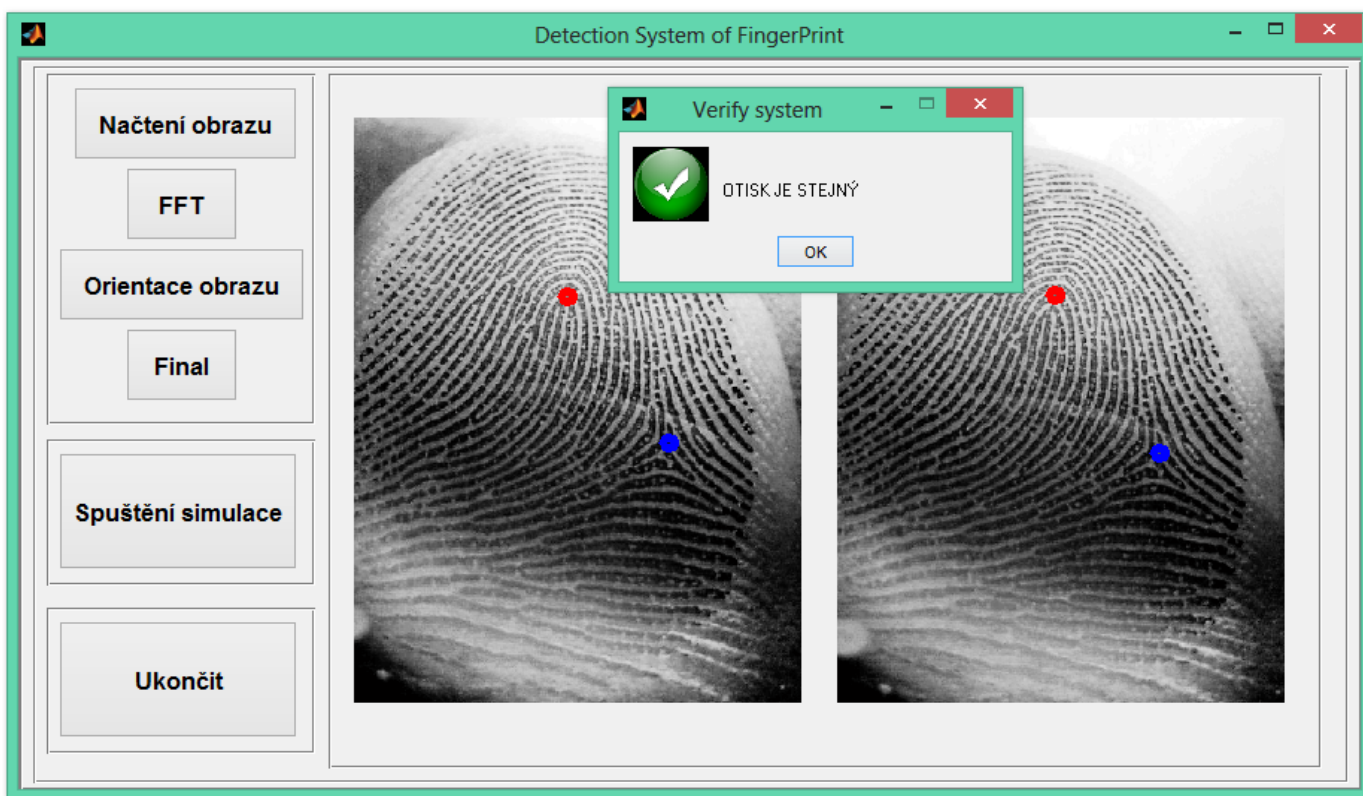


Obr. 32: Uživatelské rozhraní GUI, typ prstu – pravá smyčka

4.2 Srovnání otisků prstů – levá smyčka (left loop)

Stejně jako v předchozím případě obsahuje otisk prstu jeden singulární bod “jednoduché” **jádro** a jeden singulární bod **delta**, ale singulární bod **delta** se nachází na pravé straně a dále dochází k natočení smyčky doleva, díky tomu se jedná o otisk typu **levá smyčka** (left loop).

U tohoto příkladu, algoritmus vyhodnotil, že rozdíl vzdáleností singulárního bodu “jednoduché” **jádro** a singulárního bodu **delta** je 13 pixelů, což při maximálním možném rozdílu 20 pixelů už není zanedbatelné. Jelikož rozdíl vzdálenosti větší, než 5 pixelů a menší, než 20 pixelů ($5 < A < 20$) není zanedbatelný (do 10 pixelů je zanedbatelné, od 10 – 20 pixelů – malá chyba) je nutné vypočítat korelační koeficient směrového pole. Výsledek korelačního koeficientu je 62,86 %, což je větší než 40,55 %. Rozdíl úhlů jednotlivých otisků prstů činí 0.7757, maximální možný rozdíl je 8 stupňů. Jestliže byly tyto předpoklady splněny, program vyhodnotí porovnávané otisky prstů jako shodné, viz.: obr. 33.

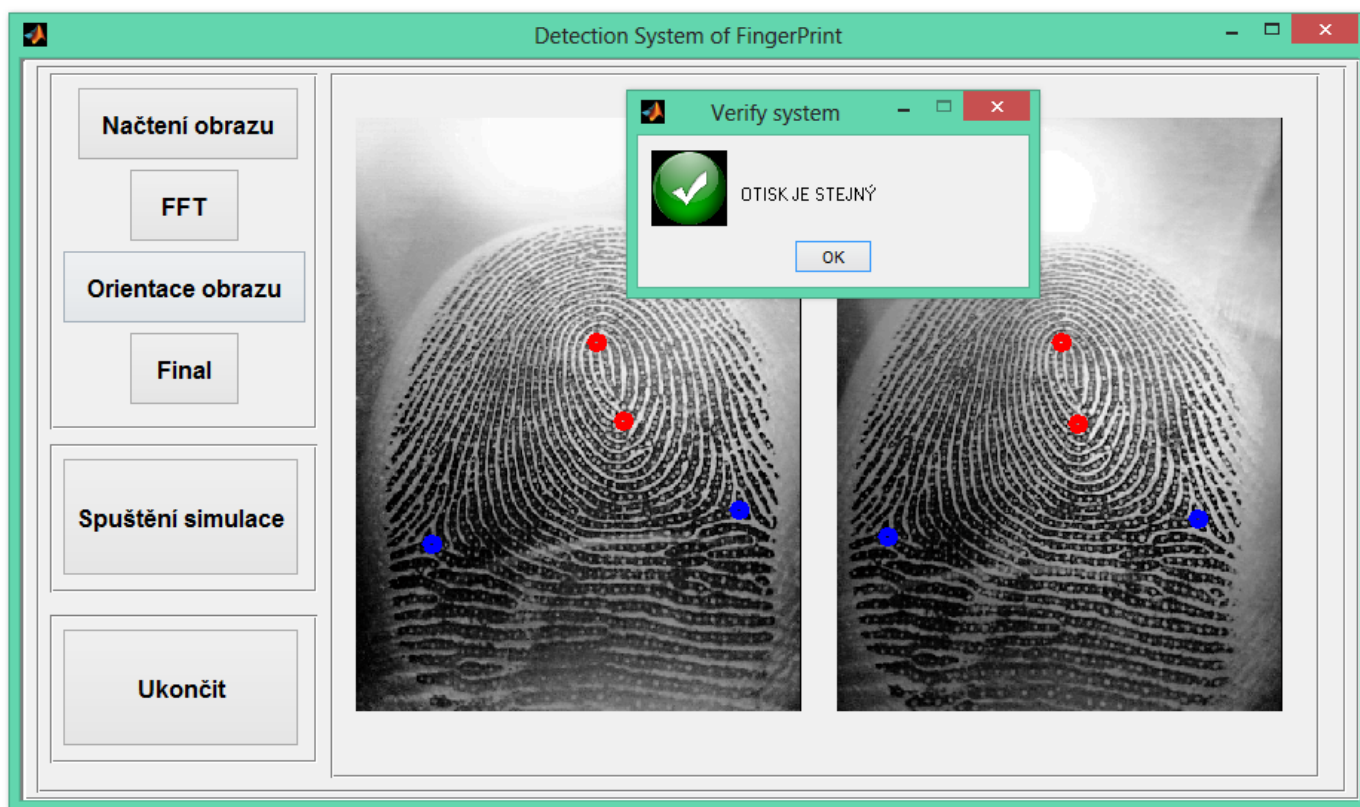


Obr. 33: Uživatelské rozhraní GUI, typ prstu – levá smyčka

4.3 Srovnání otisků prstů – závit (whorl)

V tomto případě se nejedná o otisk typu **pravá smyčka** (right loop), ale o otisk typu **závit** (whorl), a to protože daný otisk obsahuje čtyři singulární body, dva body **delta** a “**dvojité**” **jádro** (singulární bod pro horní a spodní smyčku).

U tohoto příkladu, algoritmus vyhodnotil, že rozdíl vzdáleností singulárního bodu “**dvojité**” **jádro** a singulárního bodu **delta** je 6 pixelů, což při maximálním možném rozdílu 20 pixelů už není zanedbatelné. Jelikož rozdíl vzdálenosti větší, než 5 pixelů a menší, než 20 pixelů ($5 < A < 20$) není zanedbatelný (do 10 pixelů je zanedbatelné, od 10 – 20 pixelů – malá chyba) je nutné vypočítat korelační koeficient směrového pole. Výsledek korelačního koeficientu je 45,91 %, což je větší než 40,55 %. Rozdíl úhlů jednotlivých otisků prstů činí 0. 9183, maximální možný rozdíl je 8 stupňů. Jestliže byly tyto předpoklady splněny, program vyhodnotí porovnávané otisky prstů jako shodné, viz.: obr. 34.



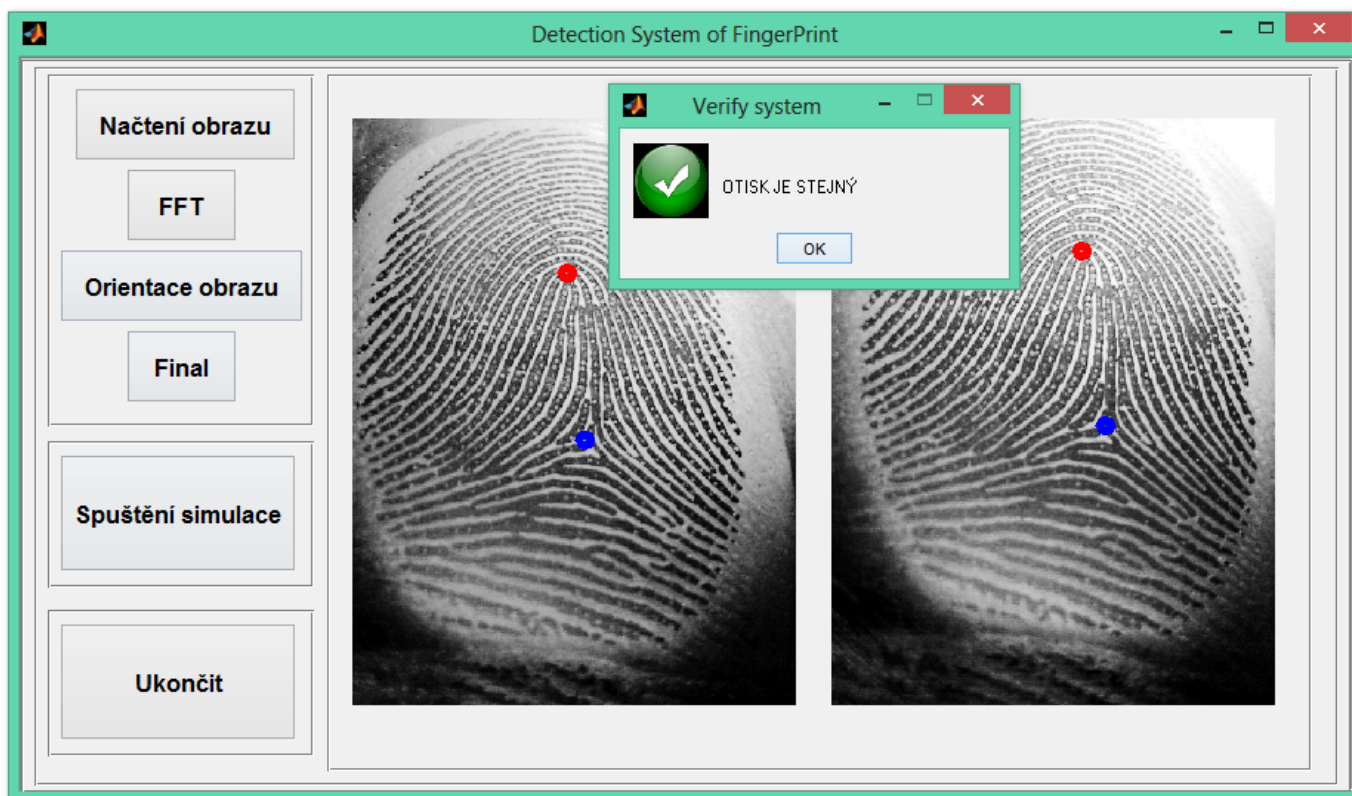
Obr. 34: Uživatelské rozhraní GUI, typ prstu - závit

4.4 Srovnání otisků prstů – klenutý oblouk (tented arch)

Porovnávané otisky prstu byly nasnímány z prstu – ukazovák. Podobně jako v předchozích případech obsahuje otisk prstu jeden singulární bod “jednoduché” **jádro** a jeden singulární bod **delta**.

Tento otisk prstu je odlišný od všech předešlých typu otisků a to v tom, že singulární bod “jednoduché” **jádro** a singulární bod **delta** mají mezi sebou úhel téměř 90° , což znamená, že leží pod sebou skoro rovnoběžně.

U tohoto příkladu, algoritmus vyhodnotil, že rozdíl vzdáleností singulárního bodu “jednoduché” **jádro** a singulárního bodu **delta** je 9 pixelů. Jelikož rozdíl vzdáleností singulárních bodů je větší, než 5 pixelů, ale menší než 20, je nutné opět jako v předchozím případě, vypočítat korelační koeficient směrového pole otisků prstů. Výsledek korelačního koeficientu se rovná 66,42 %, což je větší, než limit 40,55 %. Rozdíl úhlů jednotlivých otisků prstů činí 1.7971, maximální možný rozdíl je 8 stupňů, stejně jako u předchozího příkladu, tento předpoklad byl splněn. Jestliže byly tyto předpoklady splněny, program vyhodnotí porovnávané otisky prstů jako shodné, viz.: obr. 35.

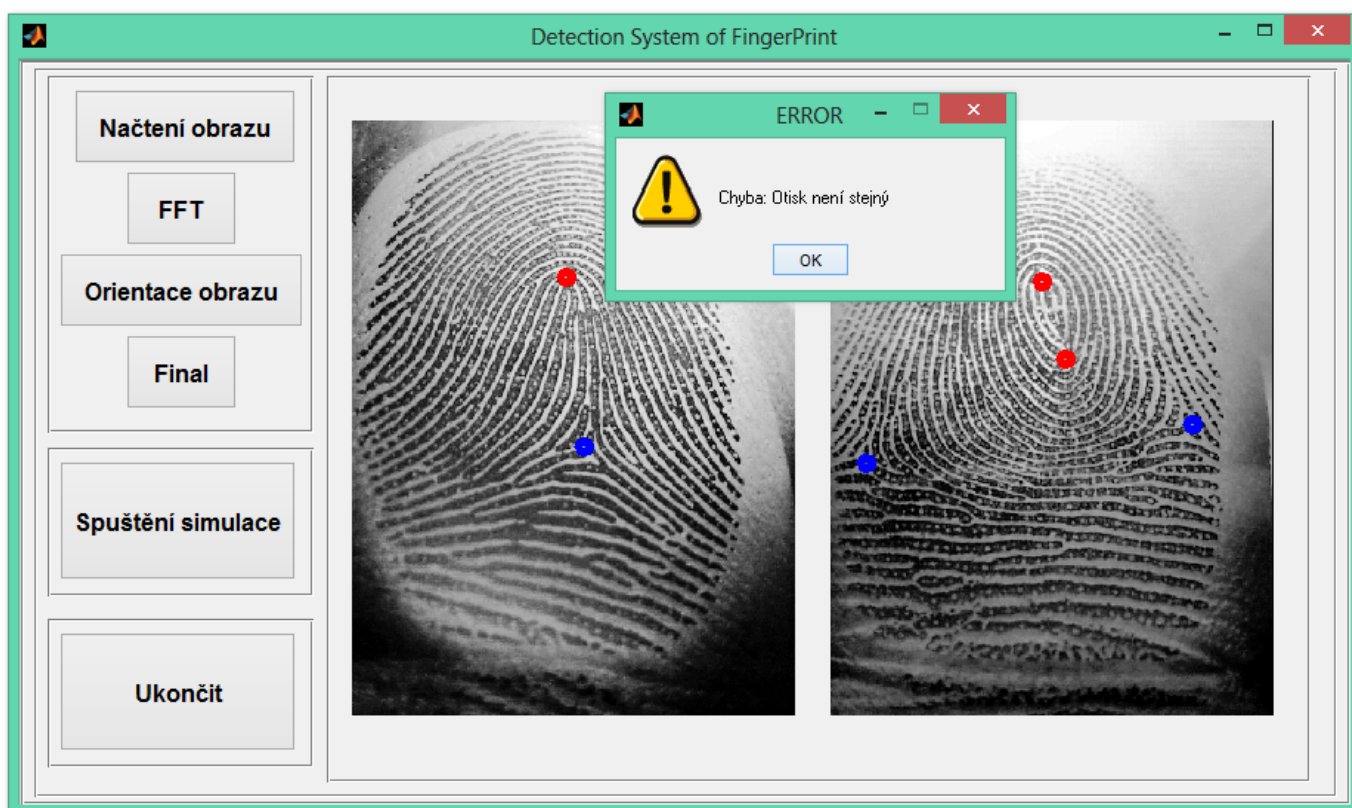


Obr. 35: Uživatelské rozhraní GUI, typ prstu – klenutý oblouk

4.5 Srovnání otisků prstů – chybové hlášení

Aby bylo zjevné, že se jedná o dva naprosto odlišné otisky prstů, tak se vybrali dva různé typy otisky otisků a to klenutý oblouk (tented arch) a závit (whorl).

U tohoto příkladu, algoritmus vyhodnotil, že rozdíl vzdáleností singulárního bodu “jednoduché” jádro a singulárního bodu δ je 24 pixelů, což při maximálním možném rozdílu 20 pixelů, je nedostačující. Rozdíl úhlů jednotlivých otisků prstů činí 34.3518, maximální možný rozdíl je 8 stupňů, stejně jako u rozdílu vzdálenosti, ale o hodně s větším (3x větší) rozdílem nebyl tento předpoklad splněn. Jestliže nebyly tyto předpoklady splněny, program vyhodnotí porovnávané otisky prstů jako falešné, viz.: obr. 36.



Obr. 36: Uživatelské rozhraní GUI, typ prstu – klenutý oblouk a závit

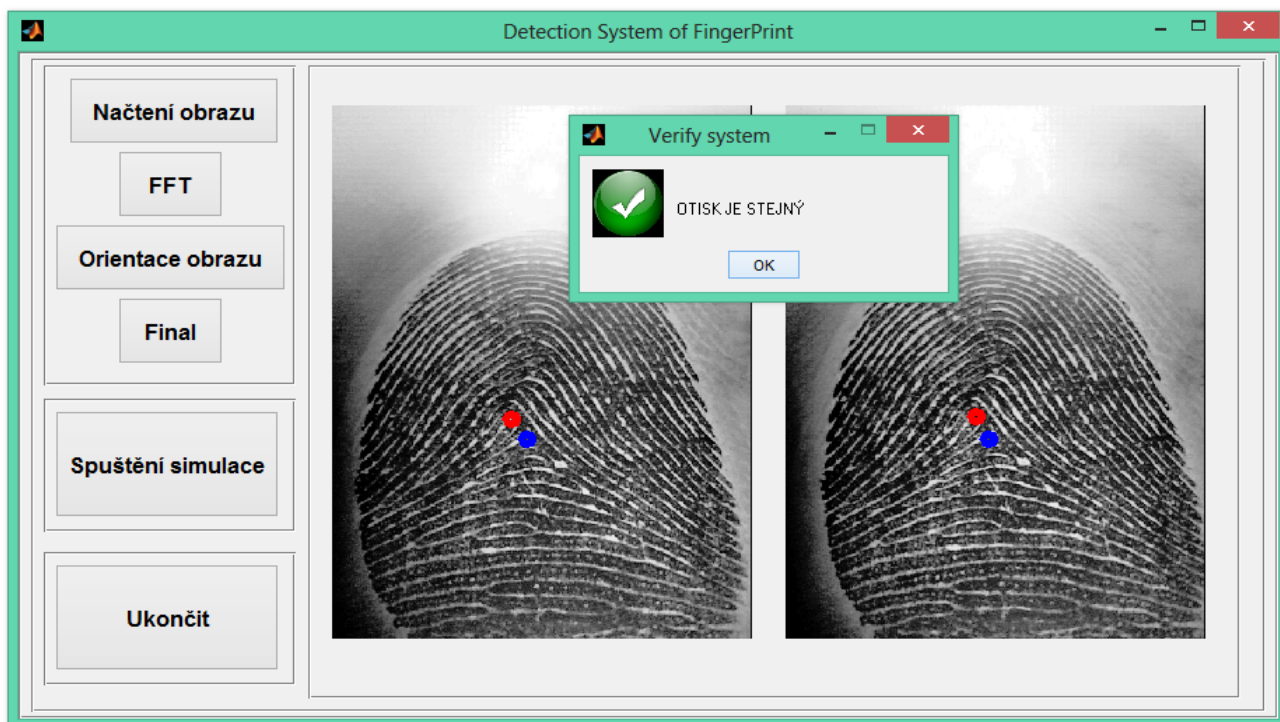
5. Zhodnocení dosažených výsledků

V této kapitole bude provedena verifikace systému pro identifikaci otisků prstů a to pomocí 10 dobrovolníků, u nichž se nasnímá vzorový otisk prstu a následně tento vzorový otisk prstu se bude porovnávat s nově nasnímaným otiskem prstu. Jednotlivé dosažené výsledky se nakonec napíší do tabulky, která bude obsahovat, o jaký typ otisku prstu se jedná, čas (určuje, jak dlouho trvalo vyhodnocení bez zobrazení jednotlivých obrazů otisků prstů), pozice x-ových a y-ových souřadnic jednotlivých singulárních bodů, vzdálenost mezi singulárními body a procentuální shodu. Na konci kapitoly bude popsáno celkové zhodnocení dosažených výsledků a to pomocí tabulky, na jejímž konci bude vypočtena celková procentuální shoda všech 10 dobrovolníků, FAR a FRR.

5.1 Verifikace systému pro identifikaci otisků prstů – dobrovolník L. P.

Tabulka 4: Výstupní hodnoty identifikačního systému po verifikaci pro dobrovolníka L. P.

Otisk	Iniciály dobrovolníka	Typ	Čas [s]	Pozice [x,y]		Vzdálenost [Pixel]	Úhel [°]	Shoda [%]
				Delta	"Jednoduché" jádro			
Vzor	L.P.	Klenutý oblouk	0,87	[225, 401]	[207, 377]	30	32,352	99
Porovnáváný				[234, 402]	[220, 374]	31	25,005	

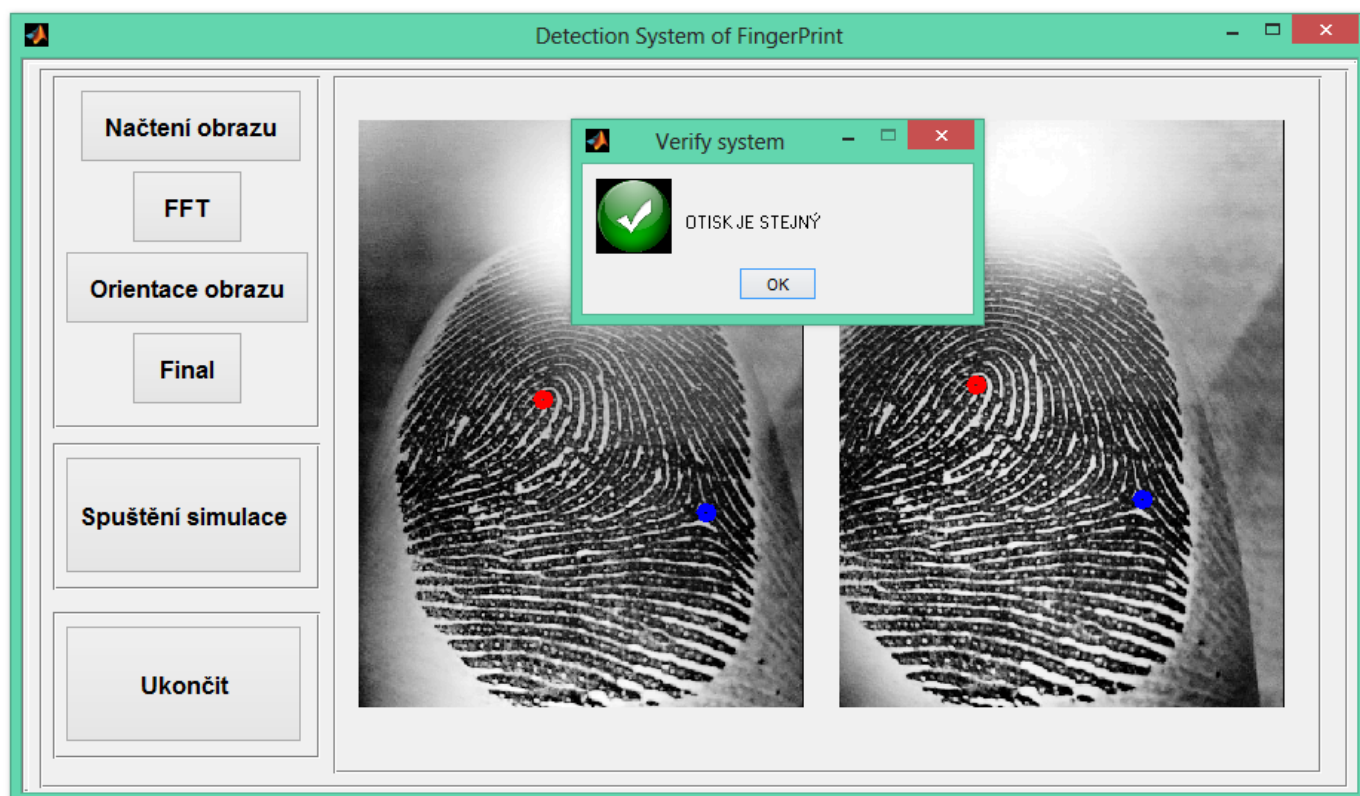


Obr. 37: Uživatelské rozhraní GUI, typ prstu – klenutý oblouk

5.2 Verifikace systému pro identifikaci otisků prstů – dobrovolník P. L.

Tabulka 5: Výstupní hodnoty identifikačního systému po verifikaci pro dobrovolníka P. L.

Otisk	Iniciály dobrovolníka	Typ	Čas [s]	Pozice [x,y]		Vzdálenost [Pixel]	Úhel [°]	Shoda [%]
				Delta	"Jednoduché" jádro			
Vzor	P.L.	Levá smyčka	0,87	[377,428]	[201,306]	214	41,986	96
Porovnáváný				[328,414]	[149,289]	218	41,935	



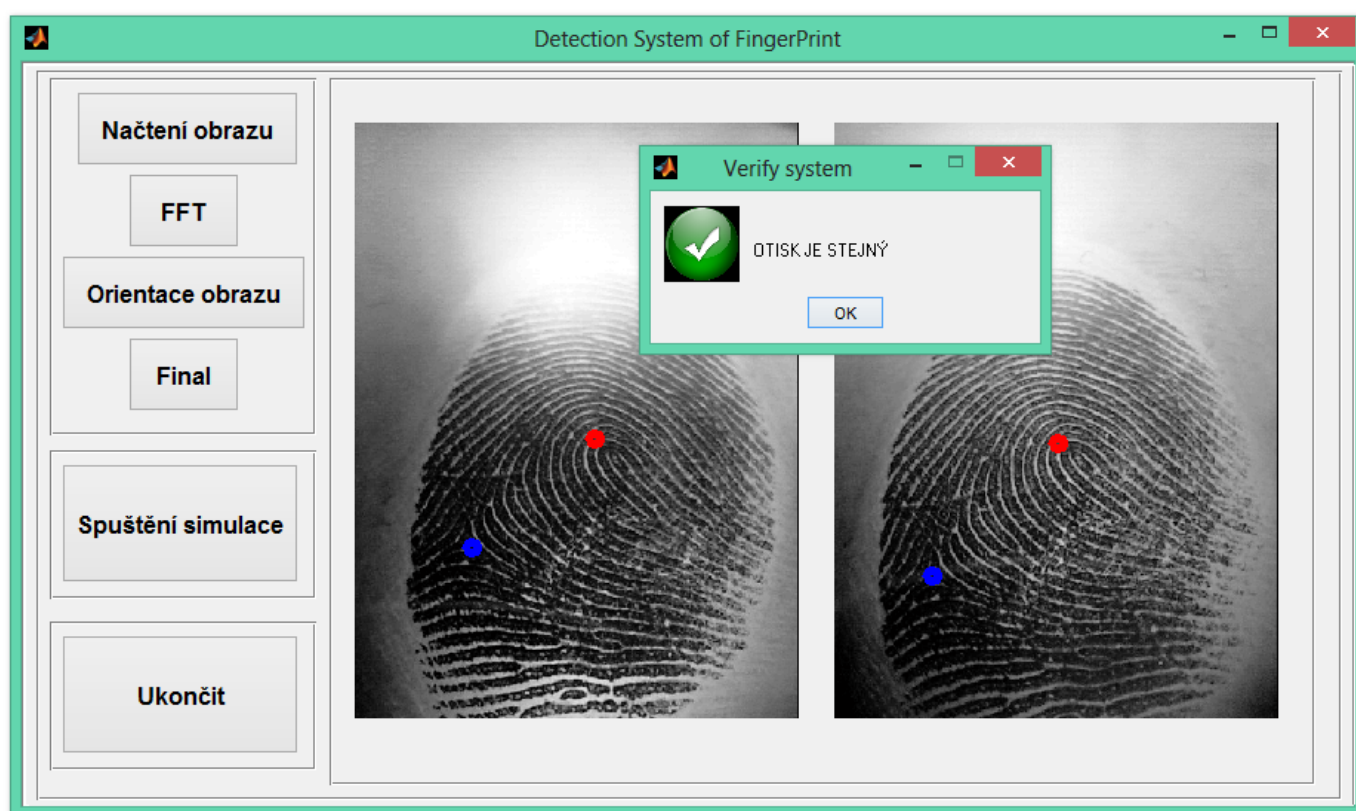
Obr. 38: Uživatelské rozhraní GUI, typ prstu – levá smyčka

5.3 Verifikace systému pro identifikaci otisků prstů – dobrovolník L. L.

Tabulka 6: Výstupní hodnoty identifikačního systému po verifikaci pro dobrovolníka L. L.

Otisk	Iniciály dobrovolníka	Typ	Čas [s]	Pozice [x,y]		Vzdálenost [Pixel]	Úhel [°]	Shoda [%]
				Delta	"Jednoduché" jádro			
Vzor	L.L.	Pravá smyčka	0,88	[128,457]	[261,341]	176	39,29	80
Porovnáváný				[107,487]	[243,346]	196	36,64	

Jelikož shoda je 80 %, což je spodní limit pro přijetí shody porovnávaných otisků a rozdíl vzdáleností je větší, než 5 pixelů, ale menší než 20 pixelů, je nutné udělat korelační koeficient směřového pole porovnávaných otisků. Korelační koeficient porovnávaných otisků prstů činí 51 %. Jelikož korelační koeficient je větší, než spodní hranice pro určení shody otisků prstů 40,55 % - jedná se o shodné otisky prstů.



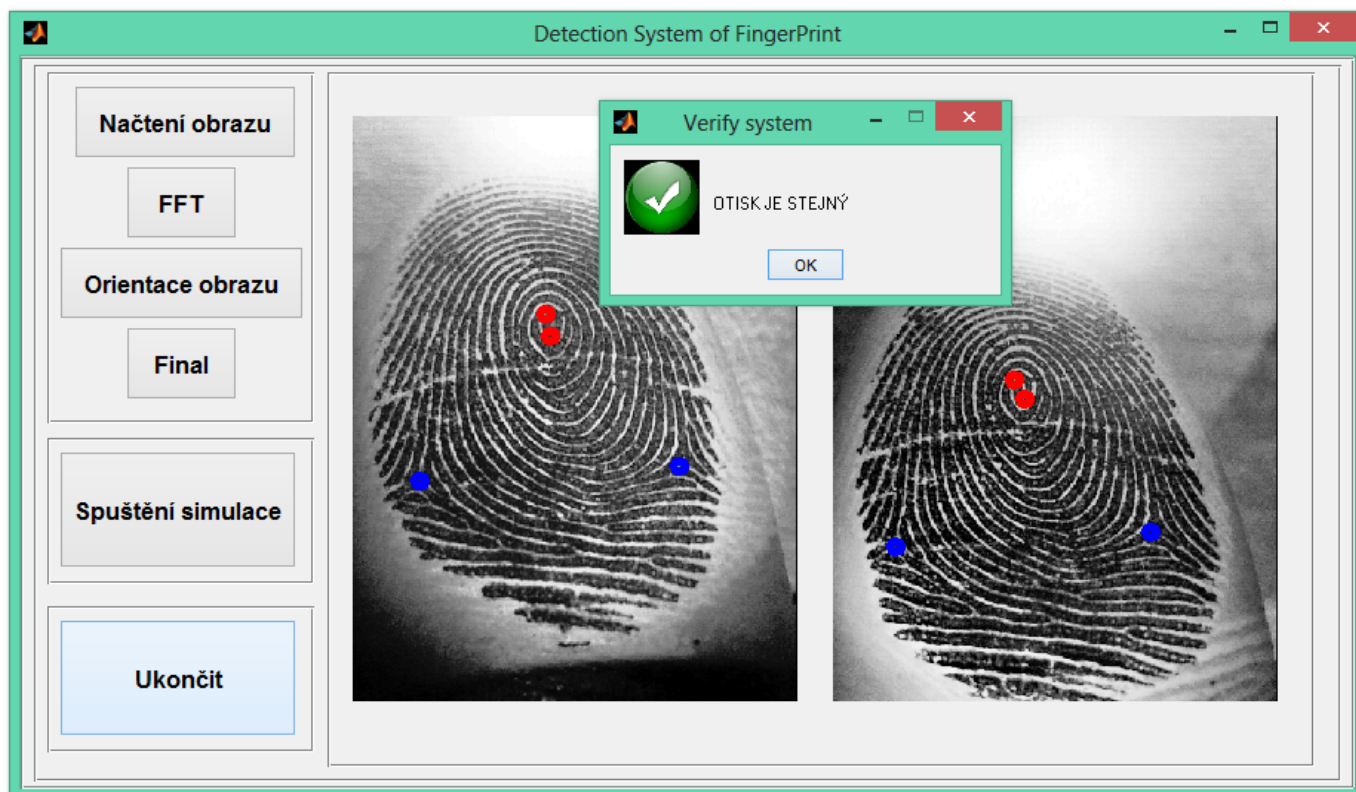
Obr. 39: Uživatelské rozhraní GUI, typ prstu – pravá smyčka

5.4 Verifikace systému pro identifikaci otisků prstů – dobrovolník M. S.

Tabulka 7: Výstupní hodnoty identifikačního systému po verifikaci pro dobrovolníka M. S.

Otisk	Iniciály dobrovolníka	Typ	Čas [s]	Pozice [x,y]		Vzdálenost [Pixel]	Úhel [°]	Shoda [%]
				Delta	"Dvojité" jádro			
Vzor	M.S.	Závit	0,85	[74,400]/[354,383]	[210,217]/[215,241]	198	35,2	98
Porovnáváný				[70,471]/[344,456]	[198,290]/[209,310]	200	35,35	

Z tabulky č. 7 si lze všimnout, že pozice pro singulární body jsou napsány dvojité. Je to hlavně proto, že se jedná o otisk typu závit, což znamená, že otisk prstu obsahuje čtyři singulární body – dva pro singulární bod **delta** a dva pro singulární bod **"dvojité" jádro** (horní a spodní smyčka). Proto každá pozice x-ových a y-ových souřadnic obsahuje dvě hranaté závorky oddělené lomítkem.

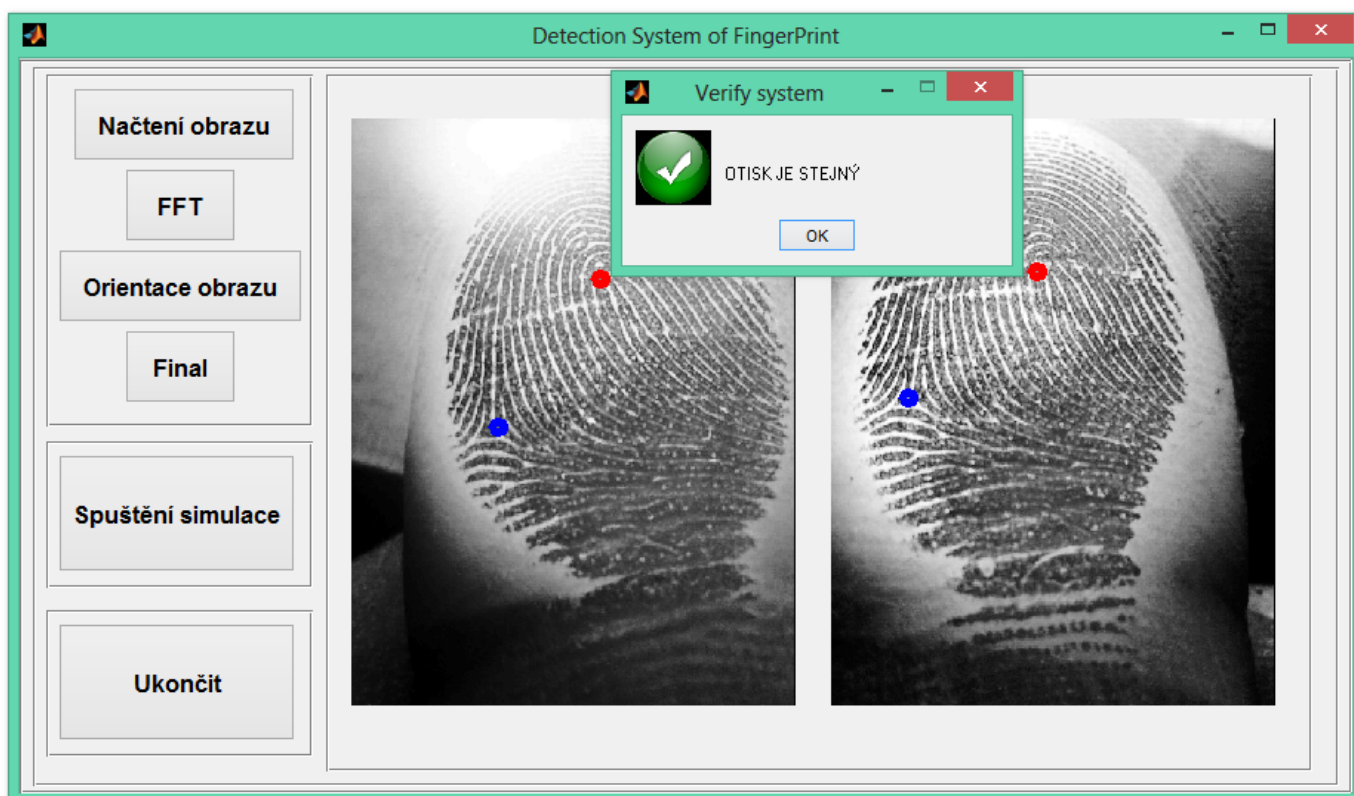


Obr. 40: Uživatelské rozhraní GUI, typ prstu – závit

5.5 Verifikace systému pro identifikaci otisků prstů – dobrovolník F. S.

Tabulka 8: Výstupní hodnoty identifikačního systému po verifikaci pro dobrovolníka F. S.

Otisk	Iniciály dobrovolníka	Typ	Čas [s]	Pozice [x,y]		Vzdálenost [Pixel]	Úhel [°]	Shoda [%]
				Delta	"Jednoduché" jádro			
Vzor	F.S.	Pravá smyčka	0,77	[161,338]	[271,176]	196	30,49	99
Porovnáváný				[86,305]	[224,167]	195	37,25	

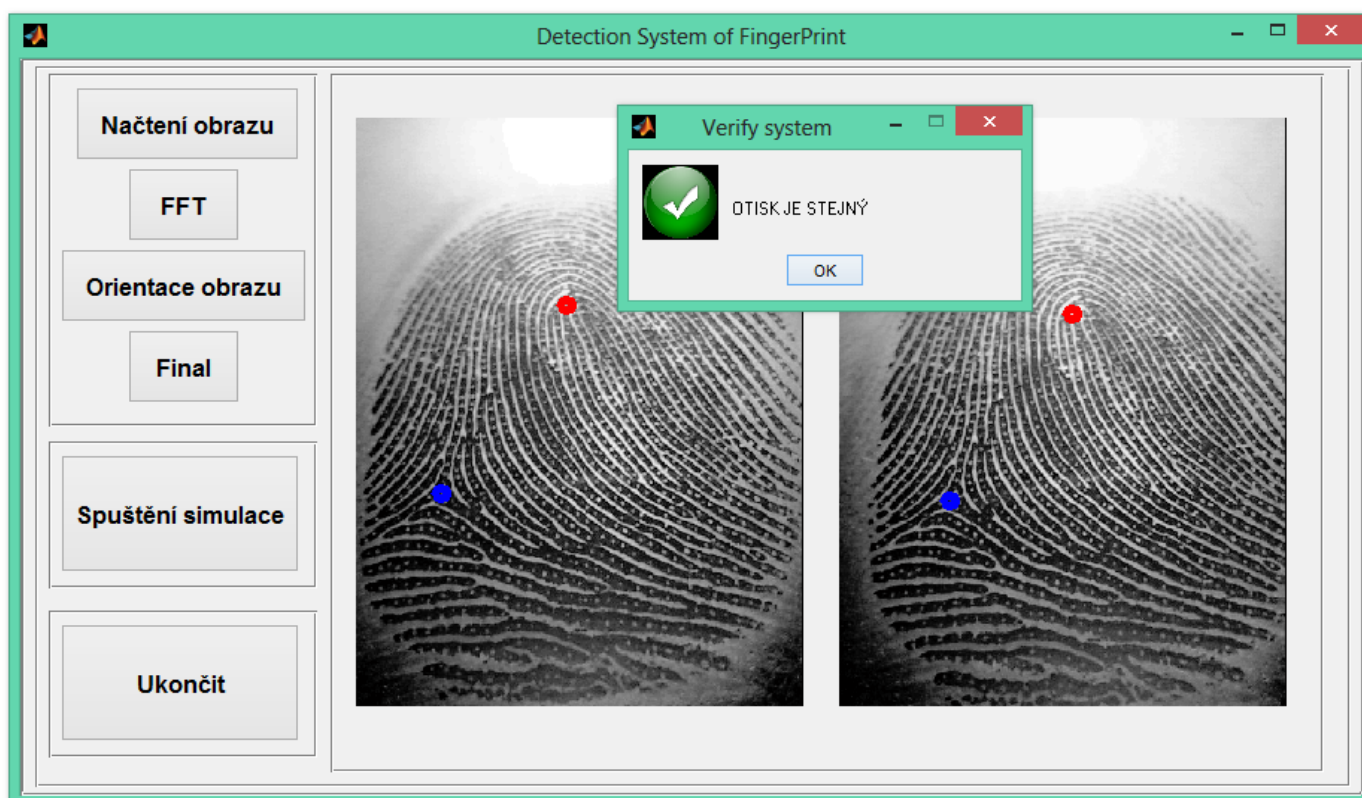


Obr. 41: Uživatelské rozhraní GUI, typ prstu – pravá smyčka

5.6 Verifikace systému pro identifikaci otisků prstů – dobrovolník L. M.

Tabulka 9: Výstupní hodnoty identifikačního systému po verifikaci pro dobrovolníka L. M.

Otisk	Iniciály dobrovolníka	Typ	Čas [s]	Pozice [x,y]		Vzdálenost [Pixel]	Úhel [°]	Shoda [%]
				Delta	"Jednoduché" jádro			
Vzor	L.M.	Pravá smyčka	0,78	[93,409]	[227,205]	244	29,91	98
Porovnáváný				[121,417]	[252,214]	242	29,52	



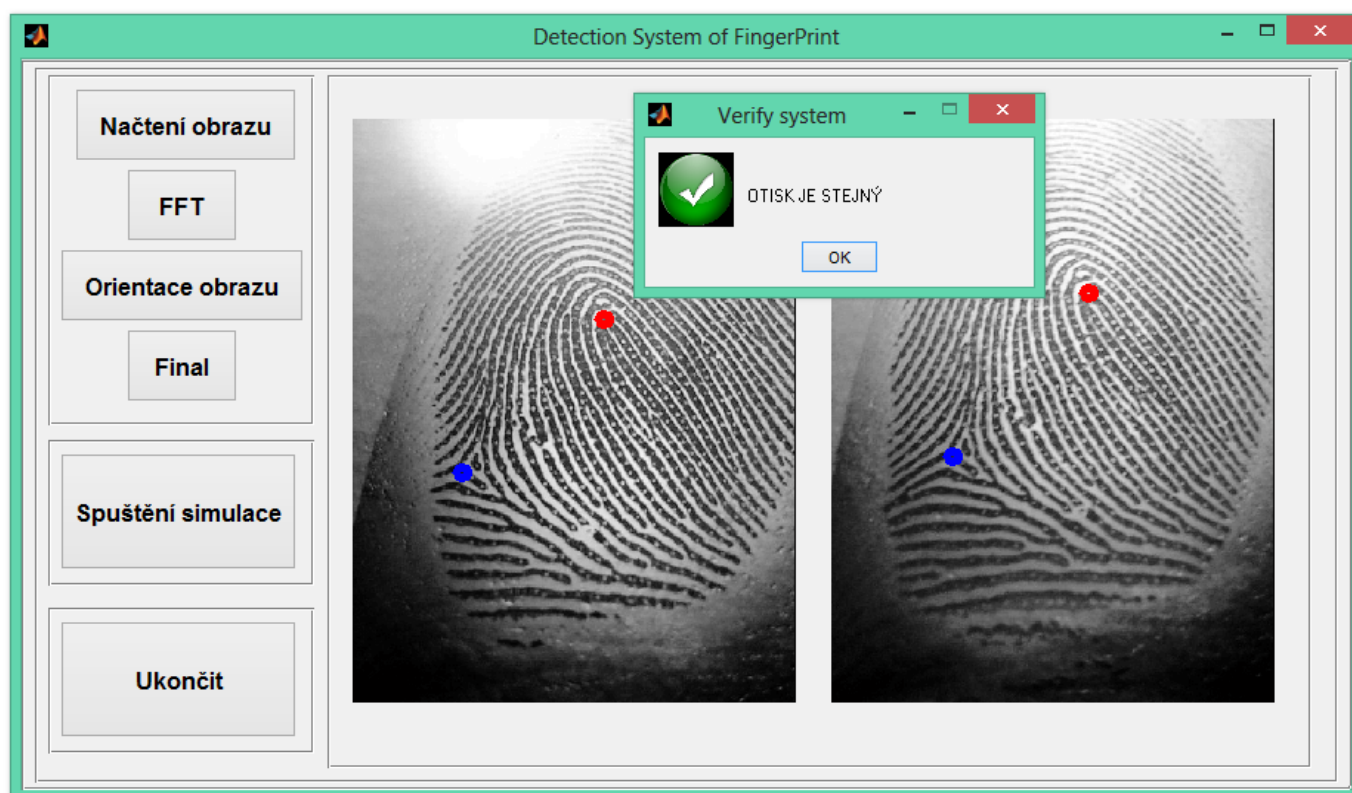
Obr. 42: Uživatelské rozhraní GUI, typ prstu – pravá smyčka

5.7 Verifikace systému pro identifikaci otisků prstů – dobrovolník M. R.

Tabulka 10: Výstupní hodnoty identifikačního systému po verifikaci pro dobrovolníka M. R.

Otisk	Iniciály dobrovolníka	Typ	Čas [s]	Pozice [x,y]		Vzdálenost [Pixel]	Úhel [°]	Shoda [%]
				Delta	"Jednoduché" jádro			
Vzor	M.R.	Pravá smyčka	0,80	[121,388]	[274,221]	226	35,89	94
Porovnávaný				[134,371]	[281,192]	242	33,92	

Jelikož rozdíl vzdáleností je větší, než 5 pixelů, ale menší než 20 pixelů, je nutné stejně jako u dobrovolníka L.L. udělat korelační koeficient směrového pole porovnávaných otisků. Korelační koeficient porovnávaných otisků prstů činí 40,60 %. Jelikož korelační koeficient je větší, než spodní hranice pro určení shody otisků prstů 40,55 % a procentuální shoda je 94 % - jedná se o shodné otisky prstů.

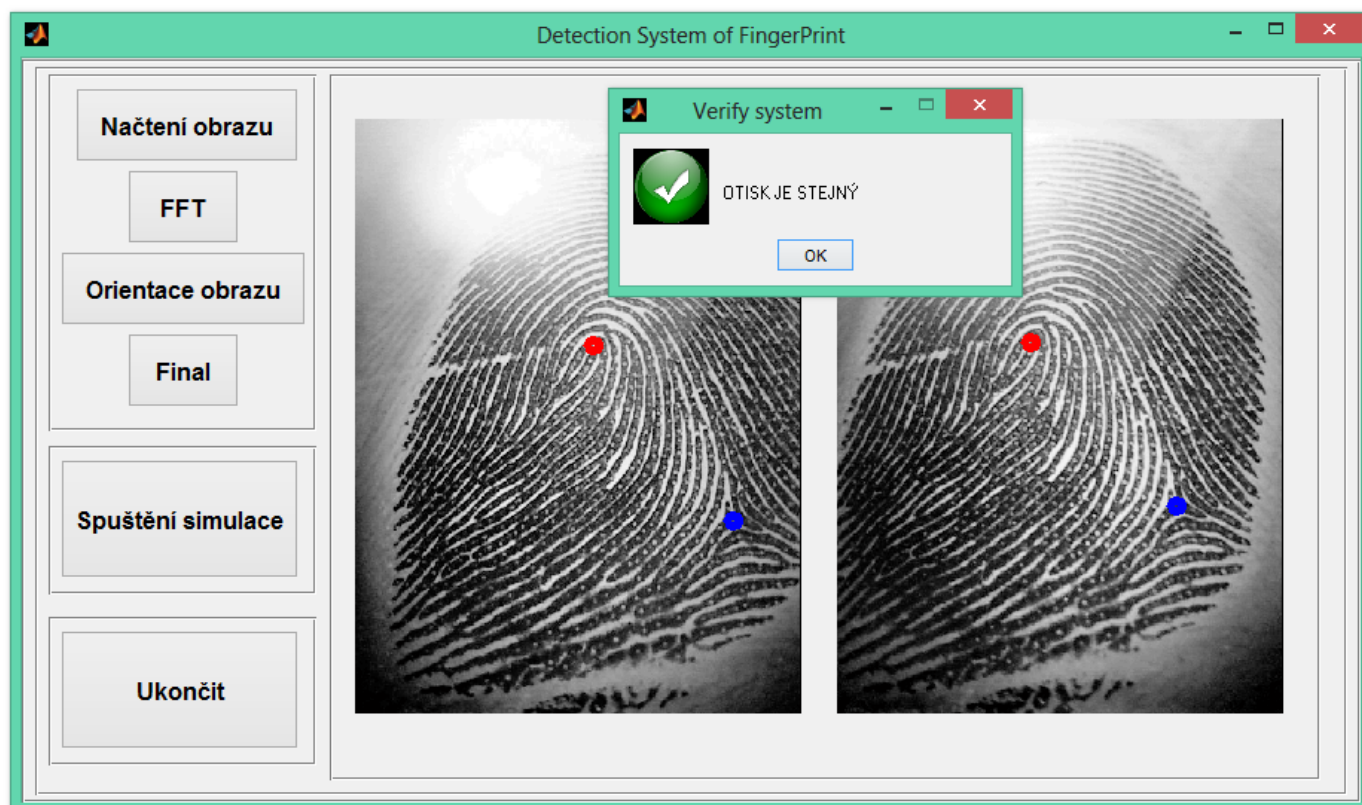


Obr. 43: Uživatelské rozhraní GUI, typ prstu – pravá smyčka

5.8 Verifikace systému pro identifikaci otisků prstů – dobrovolník D. G.

Tabulka 11: Výstupní hodnoty identifikačního systému po verifikaci pro dobrovolníka D. G.

Otisk	Iniciály dobrovolníka	Typ	Čas [s]	Pozice [x,y]		Vzdálenost [Pixel]	Úhel [°]	Shoda [%]
				Delta	"Jednoduché" jádro			
Vzor	D.G.	Levá smyčka	0,81	[408,433]	[258,245]	241	33,41	95
Porovnávaný				[367,417]	[210,241]	236	35,37	



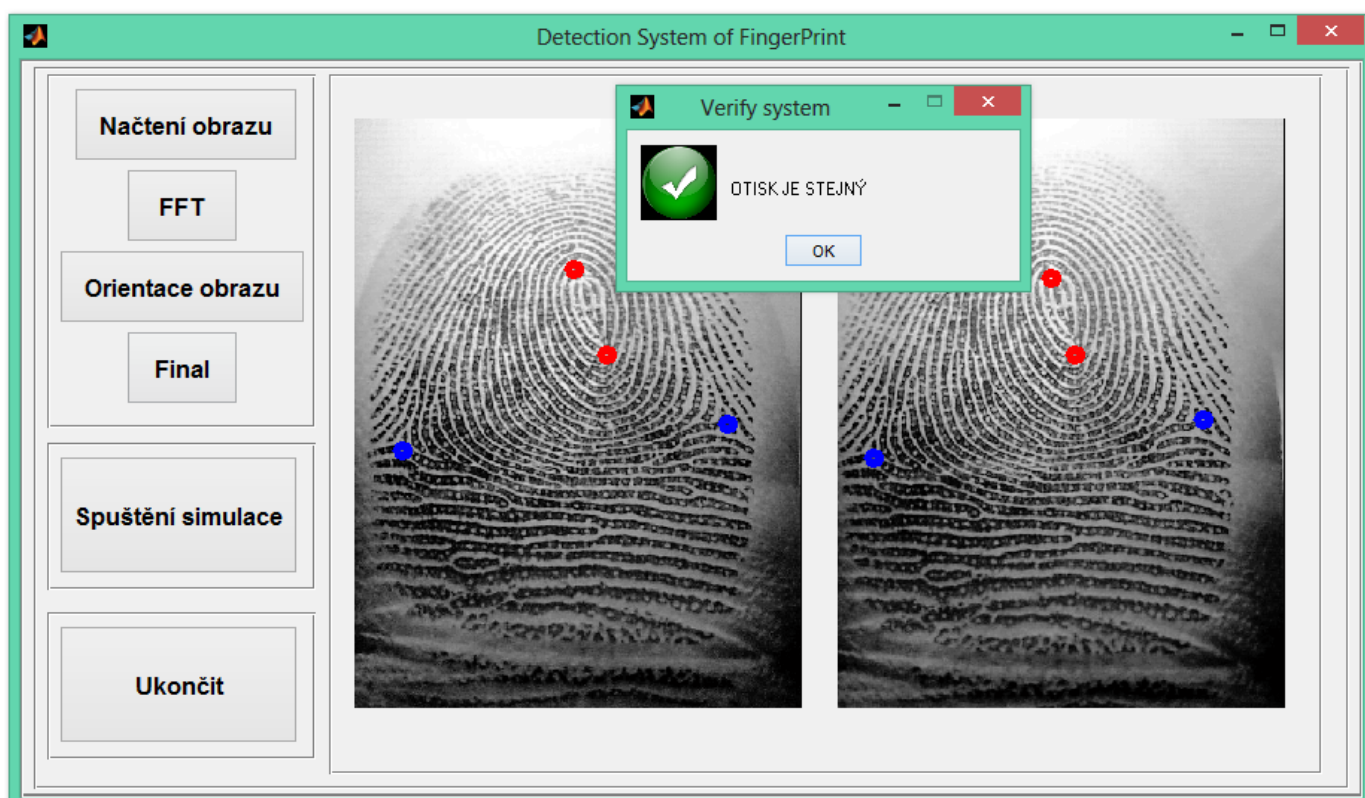
Obr. 44: Uživatelské rozhraní GUI, typ prstu – levá smyčka

5.9 Verifikace systému pro identifikaci otisků prstů – dobrovolník D. T.

Tabulka 12: Výstupní hodnoty identifikačního systému po verifikaci pro dobrovolníka D. G.

Otisk	Iniciály dobrovolníka	Typ	Čas [s]	Pozice [x,y]		Vzdálenost [Pixel]	Úhel [°]	Shoda [%]
				Delta	"Dvojité" jádro			
Vzor	D.T.	Závit	0,85	[54,361]/[402,333]	[237,164]/[273,257]	239	45,16	99
Porovnáváný				[40,370]/[374,328]	[231,174]/[256,257]	238	44,55	

Stejně jako u v bodě 5.4 u dobrovolníka M. S. se jedná o otisk typu závit, což znamená, že opět obsahuje čtyři singulární body.

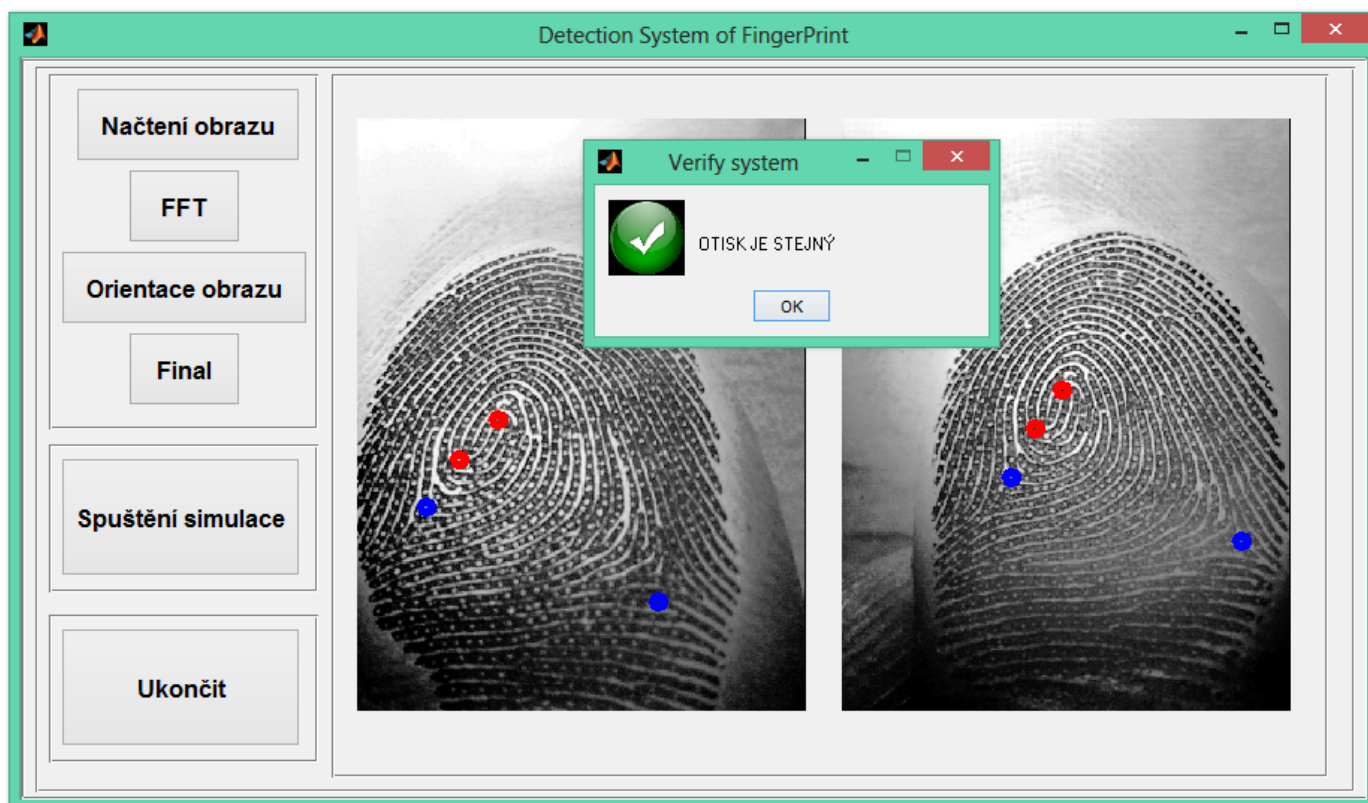


Obr. 45: Uživatelské rozhraní GUI, typ prstu – závit

5.10 Verifikace systému pro identifikaci otisků prstů – dobrovolník S. P.

Tabulka 13: Výstupní hodnoty identifikačního systému po verifikaci pro dobrovolníka S. P.

Otisk	Iniciály dobrovolníka	Typ	Čas [s]	Pozice [x,y]		Vzdálenost [Pixel]	Úhel [°]	Shoda [%]
				Delta	"Dvojité" jádro			
Vzor	S.P.	Závit	0,87	[75,420]/[323,523]	[152,326]/[111,370]	64	29,8	96
Porovnáváný				[183,388]/[430,457]	[238,294]/[209,335]	60	22,31	



Obr. 46: Uživatelské rozhraní GUI, typ prstu – závit

5.11 Celkové zhodnocení jednotlivých výsledků verifikace

V této podkapitole jsou sjednoceny všechny výsledky pro 10 dobrovolníků s tím, že je spočtena celková procentuální shoda a následně FAR (falešné přijetí osoby) a FRR (falešné odmítnutí osoby), viz.: podkapitola 1. 5.

Celková procentuální shoda je 95,4 %, což je velice dostačující hodnota. Dále je vypočtena chyba FAR, což je chyba falešného přijetí osoby. Chyba FAR se rovná 1 %, což je na hranici nízké míry spolehlivosti. Chyba FRR, což je chyba falešného odmítnutí, se rovná 5 %. Výsledky těchto dvou vypočtených chyb svědčí o tom, že systém se nehodí pro forenzní aplikace, spíše pro komerční použití, kde spíše spoléhají na to, že je velká pravděpodobnost, že se nikdo do systému nedostane na úkor toho, že je vyšší pravděpodobnost odmítnutí oprávněné osoby.

Tabulka 14: Výstupní hodnoty identifikačního systému po verifikaci

Otisk	Iniciály dobrovolníka	Typ	Čas [s]	Pozice [x,y]		Vzdálenost [Pixel]	Úhel [°]	Shoda [%]	Celková shoda [%]	FAR [%]	FRR [%]
				Delta	"Jednoduché" jádro						
Vzor	L.P.	Klenutý oblouk	0,87	[225, 401]	[207, 377]	30	32,352	99	95,4	1	5
Porovnávání				[234, 402]	[220, 374]	31	25,005				
Vzor	P.L.	Levá smyčka	0,87	[377,428]	[201,306]	214	41,986	96			
Porovnávání				[328,414]	[149,289]	218	41,935				
Vzor	L.L.	Pravá smyčka	0,88	[128,457]	[261,341]	176	39,29	80			
Porovnávání				[107,487]	[243,346]	196	36,64				
Vzor	F.S.	Pravá smyčka	0,77	[161,338]	[271,176]	196	30,49	99			
Porovnávání				[86,305]	[224,167]	195	37,25				
Vzor	L.M.	Pravá smyčka	0,78	[93,409]	[227,205]	244	29,91	98			
Porovnávání				[121,417]	[252,214]	242	29,52				
Vzor	M.R.	Pravá smyčka	0,80	[121,388]	[274,221]	226	35,89	94			
Porovnávání				[134,371]	[281,192]	242	33,92				
Vzor	D.G.	Levá smyčka	0,81	[408,433]	[258,245]	241	33,41	95			
Porovnávání				[367,417]	[210,241]	236	35,37				
Vzor	M.S.	Závit	0,85	[74,400]/[354,383]	[210,217]/[215,241]	198	35,2	98			
Porovnávání				[70,471]/[344,456]	[198,290]/[209,310]	200	35,35				
Vzor	D.T.	Závit	0,85	[54,361]/[402,333]	[237,164]/[273,257]	239	45,16	99			
Porovnávání				[40,370]/[374,328]	[231,174]/[256,257]	238	44,55				
Vzor	S.P.	Závit	0,87	[75,420]/[323,523]	[152,326]/[111,370]	64	29,8	96			
Porovnávání				[183,388]/[430,457]	[238,294]/[209,335]	60	22,31				

6. Závěr

Cílem této diplomové práce bylo vytvořit detekční systém s identifikací osob z obrazového signálu analýzy prstu. Detekční systém je vytvořen v prostředí MATLAB a je přístupný pomocí vytvořeného uživatelského rozhraní, v němž je možné porovnávat jednotlivé obrazy otisků prstů.

Základním úkolem bylo nastudování podkladů jednotlivých algoritmů a metod pro rozpoznání otisků prstů. Na základě splnění této podmínky se přešlo k realizaci programu v prostředí MATLAB. Detekční systém byl realizován podle algoritmů a metod, které se zakládali na detekci a extrakci singulárních bodů. Podle těchto singulárních bodů se následně rozhodlo, zdali nasnímané otisky jsou stejné nebo ne.

Po realizaci v programu MATLAB bylo provedeno ověření funkčnosti detekčního systému. Funkčnost systému byla ověřena prakticky a to pomocí nasnímaných otisků prstů. Otisky prstů byly nasnímany pomocí kontaktního optického snímače, který obsahoval lampu s matným sklem (pro lepší distribuci světla), zakřivené sklo (měl se použít optický hranol, ale ten se bohužel nesehnal) a webová kamera.

Dále byla provedena verifikace a to pomocí 10 dobrovolníků, u nichž se nasníмали vzorové a porovnávané otisky prstů. Tyto otisky prstů se mezi sebou porovnávali a jednotlivé výsledky byly uvedeny do tabulky, viz.: tabulka č. 14.

Z jednotlivých výsledků verifikace je možné vyvodit, že verifikace detekčního systému proběhla úspěšně – celková procentuální shoda činí 95,6 %. Dále chyba FAR činí 1% a FRR 5 %. Výsledky těchto dvou vypočtených chyb svědčí o tom, že systém se nehodí pro forenzní aplikace, spíše tam, kde se spoléhají na to, že je velká pravděpodobnost, že se nikdo do systému nedostane na úkor toho, že je vyšší pravděpodobnost odmítnutí oprávněné osoby. Důvodem, proč hodnoty chyb FAR a FRR, jsou takové, jaké jsou, je, protože ke snímání otisků prstů nebyly použity správné pomůcky a to hlavně optický hranol, díky kterému by kvalita a efektivita systému řádově vzrostla.

Použitá literatura

- [1] ŠČUREK. *Biometrické metody identifikace osob v bezpečnostní praxi*. [online]. 2008, 57 s. [cit. 2014-11-22]. Kapitola 1: Biometrie a základní pojmy, s. 6-7. Dostupné z : http://www.fbi.vsb.cz/export/sites/fbi/040/.content/sys-cs/resource/PDF/biometricke_metody.pdf
- [2] KAZÍK, Martin. *Zpracování otisků prstu* [online]. Brno, 2011 [cit. 2014-11-22]. 49 s. Kapitola 1: Teoretická část, s. 2. Diplomová práce. Vysoké učení technické v Brně. Vedoucí práce Ing. Kamil Říha, Ph.D. Dostupné z: <https://dspace.vutbr.cz/xmlui/bitstream/handle/11012/3847/Diplomov%C3%A1%20pr%C3%A1ce.pdf?sequence=1&isAllowed=y>
- [3] Daktyloskopie. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2014-11-22]. Dostupné z: <http://cs.wikipedia.org/wiki/Daktyloskopie>
- [4] RAK, Roman. *Biometrie a identita člověka ve forenzních a komerčních aplikacích*. 1. vyd. Praha: Grada, 2008, 631 s., 32 s. barev. obr. příl. ISBN 978-80-247-2365-5. Kapitola 7: Otisky prstů, daktyloskopie – forenzní teorie a praxe, s. 157-170
- [5] Biometrie. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2014-11-23]. Dostupné z: <http://cs.wikipedia.org/wiki/Biometrie>
- [6] SMÉKAL, Ondřej. *BIOMETRIE OTISKU PRSTU* [online]. Brno, 2010, 42 s. [cit. 2014-11-23]. Bakalářská práce. Vysoké učení technické v Brně. Vedoucí práce Ing. Petr Fedra. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=28074
- [7] FILLA, David. *Biometrie otisku prstu* [online]. Brno, 2011 [cit. 2014-11-24]. 69 l. Diplomová práce. Vysoké učení technické. Vedoucí práce Ing. Petr Fedra. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=37981
- [8] Slezák, Jakub. *Senzory snímání otisků prstů* [online]. Brno, 2010 [cit. 2014-11-24]. 33 l. Diplomová práce. Vysoké učení technické. Vedoucí práce Ing. Petr Fedra. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=28640
- [9] POLÁČKOVÁ, Zuzana. *Rešerše algoritmů pro snímání a zpracování otisku prstů* [online]. Praha, 2008 [cit. 2014-12-01]. Bakalářská práce. České vysoké učení technické v Praze. Vedoucí práce Ing. Martin Komárek. Dostupné z: https://dip.felk.cvut.cz/browse/pdfcache/polacz1_2008bach.pdf
- [10] ONDRŮŠEK, Roman. *Identifikační biometrické prostředky* [online]. Zlín, 2006 [cit. 2014-12-03]. Bakalářská práce. Univerzita Tomáše Bati ve Zlíně. Vedoucí práce Ing. Mgr. Milan Kvasnica, CSc. Dostupné z: https://dspace.k.utb.cz/bitstream/handle/10563/743/ondr%C5%AF%C5%A1ek_2006_bp.pdf?sequence=1

- [11] TALANDOVÁ, Hana. *Studie využití biometrických systémů v průmyslu komerční bezpečnosti* [online]. Zlín, 2010 [cit. 2014-12-03]. Bakalářská práce. Univerzita Tomáše Bati ve Zlíně. Vedoucí práce Ing. Petr Navrátil, Ph.D. Dostupné z: https://dspace.k.utb.cz/bitstream/handle/10563/13364/talandov%C3%A1_2010_bp.pdf?sequence=1
- [12] Supported fingerprint scanners and sensors. *Neurotechnology* [online]. [cit. 2014-12-07]. Dostupné z: <http://www.neurotechnology.com/cgi-bin/fingerprint-scanners.cgi>
- [13] BARHAM, Zain. *Fingerprint recognition using matlab* [online]. 2011 [cit. 2014-12-18]. Dostupné z: <http://eng.najah.edu/sites/eng.najah.edu/files/fingerprintrecognition.pdf>. Vedoucí práce Dr. Allam Mousa.
- [14] BOVIK, Alan C. [i]Handbook of Image and Video Processing. [i]1.vyd. San Diego: Academic Press, c2000. 891 s. ISBN 0-12-119790-5. Kapitola: Basic Image Processing Techniques, s. 21 – 37.
- [15] Jiya Tian; Yanqin Peng, "Research of the Matlab application in the fingerprint identification system," *Image Analysis and Signal Processing (IASP), 2012 International Conference on*, vol., no., pp.1,5, 9-11 Nov. 2012. DOI: 10.1109/IASP.2012.6425005. Dostupné z: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6425005>
- [16] HLAVÁČ, Václav. *Zpracování signálů a obrazů*. 1. vyd. Praha: Vydavatelství ČVUT, 2001, 220 s. ISBN 80-010-2114-9. Kapitola 9: Geometrické transformace a transformace hodnot jasu, s. 143.
- [17] HALOUNOVÁ, Lena. *Zpracování obrazových dat*. Vyd. 1. V Praze: České vysoké učení technické, 2009, 102 s. ISBN 978-80-01-04253-3. Kapitola 4: Zvýraznění obrazu, s. 56 – 57.
- [18] Bo, Jin, Tang Hua Ping, and Xu Ming Lan. "Fingerprint singular point detection algorithm by Poincaré index." *wseas transactions on systems* 7.12 (2008): 1453-1462.
- [19] MATLAB. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-03-27]. Dostupné z: <http://cs.wikipedia.org/wiki/MATLAB>
- [20] *MATLAB 2003: sborník příspěvků 11. ročníku konference : Kongresové centrum ČVUT, Praha, 25.11., 2003* /. Humusoft: , 2003, 2 sv. ISBN 80-708-0526-9. Kapitola: Využití grafického uživatelského rozhraní matlabu ve výzkumu a výuce měření. Dostupné z: http://dsp.vscht.cz/konference_matlab/matlab03/blaska.pdf
- [21] PATO, Joseph N a Lynette I MILLETT. *Biometric recognition: challenges and opportunities* [online]. Washington, D.C.: National Academies Press, c2010 [cit. 2014-10-01]. xv, 165 p. ISBN 03-091-4207-5.

- [22] PHILIPS, Dwayne. [i]Image processing in C.[/i] Lawrence Kansas: R & D Publications, 1994. ISBN 0-13-104548-2. 2. digitalizované vydání v roce 2000. Kapitola: Basic edge detection, s. 47 – 52.
- [23] Kovesi, Peter. *MATLAB and OCTAVE Functions for Computer Vision and Image Processing*. Dostupné z: <http://www.csse.uwa.edu.au/~pk/research/matlabfns/FingerPrints/ridgeorient.m>

Přílohy

Zdrojový kód programu, s nasnímanými otisky prstu a diplomovou prací uloženou ve formátu “.pdf”, bude dodán v elektronické podobě na CD.